WEST

Help | Logout | Interrupt

Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers | Preferences | Cases

Search Results -

Terms	Documents
16 and 18	0

Database:
US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L22

Refine Search

Recall Text | Clear

Search History

DATE: Tuesday, September 09, 2003 Printable Copy Create Case

Set Name side by side	**Query**	**Hit Count**	**Set Name** result set
	DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=NO; OP=OR		
L22	l6 and l8	0	L22
L21	l6 and l12	0	L21
	DB=USPT; PLUR=NO; OP=OR		
L20	l6 and l12	0	L20
L19	l8 and l6	0	L19
	DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=NO; OP=OR		
L18	L12 and l8	1	L18
L17	L12 and l6	0	L17
L16	L12 and l	117	L16
L15	L12 and l10	0	L15
L14	l12 and (Telocator adj Alphanumeric adj paging adj protocol)	11	L14
L13	l12 and (Telocator adj Alphanumeric adj paging adj protocol)	11	L13
L12	(paging adj protocol)	378	L12
L11	(Allen-Bradely adj3 protocol)	0	L11
L10	(DF1 adj protocol)	2	L10
L9	(Allen-Bradely adj DF1 adj protocol)	0	L9
L8	(ASCII adj protocol)	65	L8
L7	(allen-bradely adj dfi adj protocol)	0	L7
L6	((industr$3 adj control$4) and (remote adj3 communicat$3) and protocol and interface)	46	L6
L5	((industrail adj control$4) and (remote adj3 communicat$3) and (protocol) and interface)	0	L5
L4	((industrail adj control$4) and (remote adj3 communicat$3) and (communicat$3 adj3 protocol))	0	L4
L3	((industrail adj control$4) and (remote adj3 communicat$3) and (protocol) and (communicat$3 adj3 protocol))	0	L3
L2	((industrail adj control$4) and (remote) and (protocol) and (communicat$3 adj3 protocol))	0	L2
L1	((industrail adj control$4) and (remote adj3 communicat$3) and (protocol) and (communicat$3 adj3 protocol))	0	L1

END OF SEARCH HISTORY

WEST

Generate Collection Print

L2: Entry 15 of 40 File: USPT May 6, 2003

DOCUMENT-IDENTIFIER: US 6560235 B1
TITLE: Universal communication system

Abstract Text (1):
A universal communication system (UCS) consists of a number of communication modules that connect automation devices including controllers, sensors, actuators and input/output components to various communication networks. Each module transfers data between an automation device and the network using network-specific services and protocols. All modules present the same physical, electrical and logical interface to the automation device. Configuration data that customizes the presentation of the device and its data to the network is stored in non-volatile memory in the communication module. In this way, a single device design can support multiple network types by using appropriately configured modules.

Brief Summary Text (2):
This invention relates to industrial automation systems employing communication networks, and more particularly, to a network-independent communication module interface suitable for use in all components of the automation system.

Brief Summary Text (4):
Industrial automation systems frequently use communication networks to exchange data between devices. A multitude of industrial communication network types exist, and development of new networks is a continuing trend. These networks employ a wide variety of communication models, protocols and services. Several network types share the common goal of making similar devices interchangeable by defining a common communication model, or device profile, for specific device types.

Brief Summary Text (6):
With the introduction and market acceptance of new network types it is certain that devices will be required to support new network types during their life cycle. The current method of implementing communication interfaces does not provide the flexibility of supporting additional network types without the cost and delay of modifying the device itself.

Brief Summary Text (8):
The present invention is an architecture and interface definition for a universal communication system where various communication interfaces are implemented as interchangeable modules that insulate the industrial automation device from the specifics of the communication network in question. The interchangeable nature of the communication modules allows for a single device design that can communicate on any existing, or future, network type.

Brief Summary Text (9):
Importantly, the present invention provides an abstract data-exchange architecture that conceals the specifics of the network interface hardware, communication model, protocols and services. This approach eliminates any network dependencies of the industrial automation device.

Brief Summary Text (10):
Specifically, the present invention provides an industrial automation device with multi-network connectivity employing a network-independent core component that implements the primary function of the device in combination with an appropriately configured communication module. The core component exchanges data with the

communication module via an abstract interface. The communication module presents the data to the network according to network requirements and configuration data stored in the module. The configurable nature of the communication module allows the data to be presented in accordance with a desired device profile.

Brief Summary Text (11):
The industrial automation device transfers data to/from the communication module by placing it in interface RAM (IRAM) in the communication module. Access to the IRAM is via a non-traditional indirect interface. The communication module presents itself to the CPU in the automation device as 8 memory-mapped byte-wide registers. Access to data in IRAM is achieved by first writing the desired IRAM address into designated registers, and then either reading or writing 32-bits of data to/from four other registers. With this method, the interface IRAM can be accessed using 32-bit atomic transfers regardless of the data bus size of the CPU in the automation device and the actual data width of the IRAM. Furthermore, this interface provides a large amount of IRAM space (up to 128 Kbytes) while requiring only 8 memory locations in the automation product's address space. Since data is transferred in 32-bits at a time, data items up to 32-bits in size are transferred without interruption, eliminating the possibility of data corruption due to conflicting accesses to the same IRAM location by both the automation device and the network.

Brief Summary Text (13):
The logical interface implemented in IRAM uses an abstract data exchange model rather than a communication interface model, in this way the specifics of each network are eliminated from the interface.

Brief Summary Text (14):
The IRAM is divided into two main areas; a pre-defined control/status area and a free-format data exchange area. The industrial automation device controls and monitors the transfer of data through pre-defined IRAM locations in the control/status area. Data to be exchanged with the network is placed into the data exchange area. Fixed-configuration automation devices have an IRAM location assigned to each data item when the device is developed. The communication module configuration data maps these fixed-location data items to specific network transactions. Variable configuration automation devices rely on a software configuration tool to assign an IRAM location to each data item and pass the resulting memory-map to the automation device and to the communication module as network-specific configuration data.

Brief Summary Text (15):
The automation device initializes the data exchange interface by issuing an "open" command. After the "open" command the communication module is enabled to begin exchanging data with the network. The status of data exchange is indicated by pre-defined locations in the control/status area. The automation device sends "heartbeat" commands at regular intervals, and expects timely responses from the communication module. In this manner either party can detect failure of the other. The automation device terminates the data exchange interface and stops network data exchange by issuing a "close" command.

Brief Summary Text (17):
Thus, it is another objective of this invention to employ a generic data-exchange model excluding any network-specific elements from the communication module interface while preserving access to network-specific control/status. Furthermore, it is also the objective of this invention to ensure the integrity of the data exchange interface permitting both the automation device and the communication module to take appropriate action in the case of a failure.

Brief Summary Text (18):
Notifications of occurrences in the data exchange interface are passed between the automation device and the communication module. These notifications relate to changes in status of the automation device, communication module or network, changes in the values of data items or changes in validity of data items.

Brief Summary Text (20):
The communication module is configured by setting network protocol options and

defining mappings between data in the IRAM and specific network services. Applicable network protocols and services can be mapped to IRAM locations, values in non-volatile memory or constants. The flexibility of this configuration scheme permits the automation device data to be presented to the network in any manner, including constant data items common in many "device profiles."

Drawing Description Text (2):
FIG. 1 is a combined simplified block and schematic representation of an automation system incorporating a variety of typical automation devices with network connectivity provided by an interchangeable communication module in accordance with a preferred embodiment of the present invention;

Drawing Description Text (3):
FIG. 2 is a simplified block diagram of an automation device with network connectivity provided by the communication module illustrated in FIG. 1;

Drawing Description Text (4):
FIG. 3 is a logical block diagram of the communication module illustrated in FIG. 1;

Drawing Description Text (5):
FIG. 4 defines the signal assignment of the 26-pin universal communication system bus connector shown in FIG. 5;

Drawing Description Text (6):
FIG. 5 is an illustration of the circuit board form-factor of the interchangeable communication module illustrated in FIG. 1;

Drawing Description Text (7):
FIG. 6 is a list of codes such as used to report the status of the communication module illustrated in FIG. 1;

Drawing Description Text (8):
FIG. 7 is a list of codes such as used to report the status of a generic data exchange interface of the communication module illustrated in FIG. 1;

Drawing Description Text (9):
FIG. 8 is a list of codes such as used to report, in a generic fashion, the status of a network connected to the communication module as illustrated in FIG. 1;

Drawing Description Text (11):
FIG. 10 is a list of trigger codes which may be sent to the generic data exchange interface of the communication module illustrated in FIG. 1;

Drawing Description Text (12):
FIG. 11 is a list of event codes such as may be received from the generic data exchange interface of the communication module illustrated in FIG. 1;

Drawing Description Text (13):
FIG. 12 is a state-event-matrix that describes the reaction of the communication module illustrated in FIG. 1, in response to each of the trigger codes listed in FIG. 10 and both send and receive network transactions, according to data exchange interface status code of FIG. 7 and network status code of FIG. 8;

Drawing Description Text (15):
FIG. 14 is a simplified block diagram of a Motor Controller automation device with network connectivity in compliance with the device profile of FIG. 13 provided by a suitably configured communication module;

Drawing Description Text (17):
FIG. 16 is a simplified block diagram of a Motor Controller automation device with network connectivity in compliance with the device profile of FIG. 15 provided by a suitably configured communication module;

Drawing Description Text (19):

FIG. 18 is a simplified block diagram of a Motor Controller automation device with network connectivity in compliance with the device profile of FIG. 17 provided by a suitably configured communication module;

Detailed Description Text (2):
Referring to FIG. 1, a typical automation system is comprised of several automation devices 1 connected to a common communication network 19. The automation devices receive electrical power from independent power wiring 10 or, in some cases, by power conductors included in the communication network cable. Each automation device may have one or more outputs 8, which control actuators or operator displays, and/or one or more inputs 6, which receive feedback from sensors or operator input devices. The function of some types of automation devices 1 do not require directly connected outputs or inputs, as these devices perform their function based on data received from the network and report the results back to the network. Each automation device 1 implements connectivity to the network 19 with a respective communication module 11. The communication modules 11 completely insulate the automation devices from the details of the network 19, permitting the same automation devices 1 to be interconnected with a different type of network 19 simply by replacing the communication modules 11.

Detailed Description Text (3):
Universal Communication System (UCS) Applied to an Automation Device

Detailed Description Text (4):
Referring now to FIG. 2, there is shown an example of an automation device 1 with network 19 connectivity which combines electronic hardware and software elements 2,3,4,5,7,9 and 10 that perform the primary function of the device with a communication module 11 that implements the network interface.

Detailed Description Text (5):
A power conditioning circuit 9 in the automation device converts the externally connected power 10 to a suitable level for the other hardware components 2,3,4,5 and 7 and also supplies power to the communication module 11.

Detailed Description Text (6):
The automation device 1 has a CPU 2 that directs its operation according to a control program typically stored in ROM 3 with local data storage provided by a RAM 4. In many cases the ROM 3 and/or RAM 4 are integral to the CPU 2. The control program may examine inputs 5,6 and may control outputs 7,8 as part of its operation. The actual number and type of inputs and outputs is determined by the automation device's requirements, with only one input and one output shown in the interest of simplicity.

Detailed Description Text (7):
The components described to this point, with the exception of the communication module 11, implement the primary function of the automation device. Many automation devices are capable of operating in a stand-alone mode without network communication.

Detailed Description Text (8):
The communication module 11 is logically separate from the other components and is typically implemented on a separate, interchangeable circuit board. The automation device CPU 2 exchanges data with the communication module 11 by accessing the interface RAM (IRAM) 14 through the UCS bus 12 and the UCS interface 13. The communication module 11 has a separate CPU 15 that directs its operation according to a control program stored in Flash ROM 16 with local data storage provided by a RAM 17. The communication module CPU 15 exchanges data with the automation device 1 by accessing the IRAM 14 and interacting with the UCS interface 13.

Detailed Description Text (17):
With reference now to FIGS. 2 and 3, the IRAM 14 in the communication module 11 is organized with a pre-defined control/status area 26 and a free-format data exchange area 33.

Detailed Description Text (18):

The control/status area 26 contains data structures common to all types of communication modules that facilitate the exchange of data independent of network type. The module status area 27 reports the status of the communication module 11, indicating any fatal error preventing operation of the module by indication of the appropriate status code listed in FIG. 6. The interface status area 28 reports the condition of the data exchange interface between the automation device 1 and the communication module 11 by indication of the appropriate status code from FIG. 7. The network status area 29 indicates the status of data exchange with the network communication interface and reports communication link errors by indication of a generic status code from FIG. 8. By examination of module status 27, interface status 28 and network status 29 areas, the automation device 1 can determine the complete operational status of network data exchange interface through the communication module 11 and take proper action should a fault be reported.

Detailed Description Text (19):
Referring to FIGS. 9 and 12, there are respectively shown a state transition diagram and a state event matrix illustrating the sequence of operations carried out by the Universal Communication System of the present invention. FIG. 9 shows the possible NET_STATUS_CODE transitions for the Network Status Codes shown in FIG. 8 and the occurrences that cause the transitions. In FIG. 9, each oval, or bubble, represents a state or condition, while an arrow connecting a pair of ovals represents an event or occurrence. Thus, for example, when the interface is closed and the network interface is offline, the operating program of the system can take various actions when the interface is opened. For example, the program may detect an offline fault and proceed to the OFFLINE_FAULT state for clearing a network fault. The program may also proceed to the ONLINE state for exchanging data using nonsecure services, providing no network faults are detected. Finally, the program may proceed from the interface offline state when the interface is opened to the OFFLINE_BAD_CFG state where the interface remains offline due to a configuration fault such as an invalid or duplicate station address, an invalid baud rate, etc. From FIGS. 8, 9 and 12, the various states or conditions of the system as well as the actions available when in a given state can be determined.

Detailed Description Text (20):
The device status area 30 contains data that reports the status of the automation device to the communication module. Depending on the requirements of the specific network protocol, faults indicated in the device status area 30 may be reported to the network, or may affect operation of data exchange with the network. The trigger queue 31 provides a means for the automation device 1 to pass commands and notifications of occurrences as listed in FIG. 10 to the communication module 11. Similarly, the event queue 32 provides a means for the communication module 11 to pass responses to commands and notifications of occurrences as listed in FIG. 11 to the automation device 1. The trigger and event queues 31,32 are implemented as circular queues or first-in first-out (FIFO) buffers using established programming techniques with management fields located in the control/status block 26.

Detailed Description Text (21):
To enable data exchange with the network 19, the automation device 1 sends the OPEN trigger of FIG. 10 through the trigger queue 31. Upon processing the OPEN trigger, the communication module 11 changes the interface status 28 to OPEN, enables network data exchange and sends the OPEN event of FIG. 11 through the event queue 32.

Detailed Description Text (22):
Once data exchange has been enabled as described herein, the automation device 1 sends the HEARTBEAT trigger of FIG. 10 through the trigger queue 31 at regular intervals. Upon processing the HEARTBEAT trigger, the communication module 11 (re)starts a heartbeat fault timer and sends the HEARTBEAT event of FIG. 11 through the event queue 32 by way of response. If the communication module 11 detects expiration of the heartbeat fault timer before the next HEARTBEAT trigger is received, it changes the interface status 28 to HEARTBEAT_FAULT and disables network data exchange.

Detailed Description Text (23):
Once data exchange has been enabled as described herein, the automation device 1 is free to exchange data with the network 19 by way of the data exchange area 33 in

IRAM 14. When the automation device 1 changes a data value stored in the data exchange area 33, it sends the DATA_CHANGE trigger of FIG. 10 through the trigger queue 31. Upon processing the DATA_CHANGE trigger, the communication module 11 takes the necessary action to report the new data to the network. When the communication module 11 changes a data value stored in the data exchange area 33, it sends the DATA_CHANGE event of FIG. 11 through the event queue 32 indicating the affected part(s) of the data exchange area 33. Upon processing the DATA_CHANGE event, the automation device 1 takes the necessary action to apply the new data value(s) to its operation.

Detailed Description Text (24):
Some network types make use of secure communication services that permit identification of data that is no longer valid due to a failed or suspended service. If such failure is detected, the communication module sends the DATA_INVALID event of FIG. 11 through the event queue 32 indicating the affected part(s) of the data exchange area 33. Upon processing the DATA_INVALID event, the automation device 1 takes the necessary action to react to the invalid data.

Detailed Description Text (25):
Some network types make use of secure communication services that permit identification of data that is temporarily not updated. If such condition is detected, the communication module sends the DATA_IDLE event of FIG. 11 through the event queue 32 indicating the affected part(s) of the data exchange area 33. Upon processing the DATA_IDLE event, the automation device 1 takes the necessary action to react to the idle data.

Detailed Description Text (26):
To cease the exchange of data with the network 19, the automation device 1 sends the CLOSE trigger of FIG. 10 through the trigger queue 31. Upon processing the CLOSE trigger, the communication module 11 disables network data exchange and changes the interface status 28 to CLOSED.

Detailed Description Text (27):
As described herein, the automation device 1 exchanges data with the communication module 11 by way of an abstract data exchange interface that is secure in all respects but in no way reflects the manner of presentation of data to the network 19. Furthermore, the automation device 1 can monitor and report the status of data exchange with the network 19 by way of generic status codes without any network-specific concepts.

Detailed Description Text (29):
Referring again to FIGS. 2 and 3, the communication module 11 maps network services to the data exchange area 33 in the IRAM 14 according to configuration data stored in flash ROM 16 thus completing the data path from the automation device 1 to the network 19.

Detailed Description Text (31):
The product identity data 34 supplies network-specific information such as manufacturer, product name, version, serial number, etc., that identifies the communication module 11 on the network 19.

Detailed Description Text (32):
Many networks have optional and/or configurable protocols and/or services. Protocol options configuration 35 contains network-specific data that selects and configures the protocol options supported by the communication module 1.

Detailed Description Text (33):
The data mapping configuration 36 controls how the communication module 11 presents the data exchange area 33 to the network 19. For each supported network protocol or service, the data mapping configuration 36 defines the location in the data exchange area 33 where the corresponding data is located.

Detailed Description Text (35):
Many network protocols provide general-purpose services that provide access to a large amount of data using a data address in conjunction with the service. The data

address format is network-specific in nature, and in the case of device profiles, the address for specific data items is pre-defined. In order to support these types of protocols, the data mapping configuration 36 assigns a specific part of the data exchange area 33 to each network-specific address. In this way, any pre-defined data layout with reference to network-specific addressing can be mimicked by the communication module 11 without the necessity of the actual data layout in question being mirrored by the layout of data in the data exchange area 33.

Detailed Description Text (38):
As described herein, the communication module 11 presents automation device data located in the data exchange area 33 to the network 19 as defined by product identity, protocol options and data mapping configuration data 34,35,36 stored in flash ROM 16. The configuration data has complete control over the manner in which the automation device data is presented to the network.

Detailed Description Text (42):
The following three examples illustrate how the motor controller 59 is interfaced to three different networks by the application of three different suitably configured communication modules 11.

Detailed Description Text (47):
FIG. 14 illustrates the example motor controller 59 with a communication module 11 containing data mappings 53 that implement the device profile of FIG. 13.

Detailed Description Text (48):
Referring to FIGS. 13 and 14, when the communication module 11 receives an I/O update 41 output message 38 it processes it according to the data mappings 53 stored in the Flash ROM 16. It stores the first output message 38 component, data type BOOL, in IRAM location 0301h and the second component, data type UINT, in IRAM location 0320h. After processing the output message 38, the communication module 11 builds an input message 39 according to the data mappings 53 stored in Flash ROM 16. It sets the first input message 39 component, data type BOOL, from IRAM location 0303h and the second component, data type UINT, from IRAM location 0310h.

Detailed Description Text (49):
Thus, the communication module 11 interprets output messages 38 and generates input messages 39 defined by the device profile 37, assigning the various data components of these messages to IRAM locations specified by data mappings 53 stored in Flash ROM 16.

Detailed Description Text (50):
When a parameter access request 42,43,44,45,46,47 is received, the communication module 11 processes it according to the data mappings 53 and returns an appropriate response. A request to read Param142 results in the data stored in IRAM location 0327h, data type UINT, being returned in the response message. A request to write Param143 results in the value in the request message, data type UINT, being stored in IRAM location 0327h. A request to read Param244 results in the data stored in IRAM location 0303h, data type BOOL, to be returned in the response message. A request to write Param245 results in an error response being returned since the data mappings 53 specify that Param2 is read-only. A request to read Param346 results in the constant value 0100h, data type UINT, being returned in the response message. A request to write Param347 results in an error response being returned since the data mappings 53 specify that Param3 is a constant and therefore read-only. All other read/write requests result in appropriate error responses.

Detailed Description Text (51):
Thus, the communication module 11 responds to pre-defined parameter read/write requests, storing received data in IRAM, and generating suitable responses including constants and data from IRAM as specified by data mappings 53 stored in Flash ROM 16.

Detailed Description Text (55):
FIG. 16 illustrates the example motor controller 59 with a communication module 11 containing data mappings 58 that implement the device profile of FIG. 15.

Detailed Description Text (56):
Referring to FIGS. 15 and 16, when the communication module 11 receives an I/O update 57 output message 55 it processes it according to the data mappings 58 stored in the Flash ROM 16. It stores the first output message 55 component, data type BOOL, in IRAM location 0301h, the second component, data type UINT, in IRAM location 0320h and the third component, data type UINT, in IRAM location 327h. After processing the output message 55, the communication module builds an input message 56 according to the data mappings 58 stored in Flash ROM 16. It sets the first input message 56 component, data type BOOL, from IRAM location 0303h and the second component, data type UINT, from IRAM location 0310h.

Detailed Description Text (57):
Thus, the communication module 11 interprets output messages 55 and generates input messages 56 defined by the device profile 54, assigning the various data components of these messages to IRAM locations specified by data mappings 58 stored in Flash ROM 16.

Detailed Description Text (61):
FIG. 18 illustrates the example motor controller 59 with a communication module 11 containing data mappings 72 that implement the device profile of FIG. 17.

Detailed Description Text (62):
Referring to FIGS. 17 and 18, when a register access request 62, 63, 64, 65, 66, 67, 68, 69, 70 and 71 is received, the communication module 11 processes it according to the data mappings 72 and returns an appropriate response. A request to read register 4000162 results in the data stored in IRAM location 0320h, data type UINT, being returned in the response message. A request to write register 4000163 results in the value in the request message, data type UINT, being stored in IRAM location 0320h. A request to read register 4000264 results in the data stored in IRAM location 0301h, data type BOOL, being returned in the response message. Since the network does not support the BOOL data type, an automatic conversion to a suitable network data type is performed. A request to write register 4000265 results in the value in the request message being converted to data type BOOL and stored in IRAM location 0301h. A request to read register 4000366 results in the data stored in IRAM location 0327h, data type UINT, being returned in the response message. A request to write register 4000367 results in the value in the request message, data type UINT, being stored in IRAM location 0327h. A request to read register 4000468 results in the data stored in IRAM location 0310h, data type UINT, being returned in the response message. A request to write register 4000469 results in an error response being returned since the data mappings 72 specify that register 40004 is read-only. A request to read register 4000570 results in the data stored in IRAM location 0303h, data type BOOL, being returned in the response message. Since the network does not support the BOOL data type, an automatic conversion to a suitable network data type is performed. A request to write register 4000571 results in an error response being returned since the data mappings 72 specify that register 40005 is read-only. All other read/write requests result in appropriate error responses.

Detailed Description Text (63):
Thus, the communication module 11 responds to pre-defined register read/write requests, storing received data in IRAM, and generating suitable responses including data from IRAM as specified by data mappings 72 stored in Flash ROM 16.

Detailed Description Text (64):
There has thus been shown a universal communication system which includes a number of communication modules each of which connects an automation device such as a controller, sensor, actuator or input/output device to any one of various communication networks each having its own network-specific operating services and protocols, or procedures. All communications modules present the same physical, electrical and logical interface to a given automation device and each module is further adapted to transfer data between its associated device and the networks using network-specific services and protocols. Each communication module includes an interface random access memory (IRAM) and a non-volatile flash read only memory (ROM). The IRAM is divided into two main areas: a pre-defined control/status area and a free-format data exchange area. Pre-defined locations in the IRAM are used to control and monitor the transfer of data, while data to be exchanged with a network

is placed in the IRAM's data exchange area. Automation devices have a specific IRAM location assigned to each data item, either pre-defined by the automation device designer, or defined by variable configuration data stored in the automation device. The Flash ROM is programmed to accommodate the IRAM location assignment of the automation device by assigning an IRAM location and corresponding network transactions for each data item for use by the communication module as network-specific configuration data for any one of the several networks. Data is read from selected IRAM locations and provided to a network in accordance with the specific operating services and protocols of the requesting network.

CLAIMS:

1. A communication method between a plurality of automation devices and any one of a plurality of networks, wherein each network is characterized by a set of network-specific services and protocols and each automation device is characterized by device-specific data, said method comprising the steps of: organizing an interface RAM into a first control/status area for storing data common to the operation of all of the automation devices and a second free-format data exchange area for storing device-specific data for each of the automation devices; assigning each of the data common to the operation of all of the automation devices to a respective first data address location in the first control/status area of the interface RAM; assigning each of the device-specific data to a respective second data address location in the second free-format data exchange area of the interface RAM, wherein data is written into and read from said first and second data address locations in accordance with a specified universal service and protocol independent of each of said networks; and providing data from said first and second data address locations respectively in said first control/status area and said second free-format data exchange area to a given network in accordance with the specified service and protocol of said given network and configuration data stored in a non-volatile memory.

9. A communication system between a plurality of automation devices and any one of a plurality of networks, wherein each network is characterized by a set of network-specific services and protocols and each automation device is characterized by device-specific data, said system comprising: a random access memory having a first control/status data storage area for storing data common to the operation of all of the automation devices and a second free-format data exchange area for storing device-specific data for each of the automation devices; a controller coupled to said random access memory for assigning each of the data common to operation of all of the automation devices to a respective first data address location in said first control/status data storage area and assigning each of the device-specific data to a respective second data address location in the second free-format data exchange area of said random access memory, wherein said controller further writes data into and reads data from said first and second data address locations in accordance with a specified universal service and protocol independent of each of the networks; and an interface circuit coupling said random access memory to said plurality of networks for providing data from said first and second data address locations respectively in said first control/status area and said second free-format data exchange area to a given network in accordance with the specified service and protocol of said given network and configuration data stored in said controller.

10. The communication system of claim 9 wherein said random access memory is an interface random access memory (IRAM) and said controller is a flash read only memory (ROM), and wherein said flash ROM is programmed to assign each of the data common to operation of all of the automation devices to a respective first data address location in said first control/status data storage area and to assign each of the device-specific data to a respective second data address location in the second free-format data exchange area of said IRAM.

11. The communication system of claim 10 wherein a device-specific data for each of the automation devices includes product identity information, protocol options or data mappings.

12. The communication system of claim 11 further comprising an operating device

coupled to and controlled or monitored by said automation device.

13. The communication system of claim 12 wherein said operating device is a control actuator or video display and said communication system controls the operation of said control actuator or video display.

14. The communication system of claim 12 wherein said operating device is a sensor or operator input device and said communication system monitors outputs from said sensor or operator input device.

15. The communication system of claim 10 wherein said IRAM, flash ROM, and interface circuit comprise a communication module.

16. A communication system for coupling a plurality operating devices to a network, wherein said network is characterized by a set of network-specific services and protocols and communicates control/status information to/from each of said operating devices, and wherein said control/status information includes data common to all of the operating devices and device specific data for each of the operating devices, said communication system comprising: a plurality of communications modules each coupling a respective operating device to the network; a plurality of first memories each associated with a respective communications module, wherein each of said first memories includes a first control/status area having a first plurality of data address locations and a second free-format data exchange area having a second plurality of data address locations, wherein the data common to all of the operating devices is stored in said first plurality of data address locations; and a plurality of second memories each associated with a respective communications module and its associated first memory, wherein each of said second memories includes a data mapping arrangement for writing device-specific data into and reading device-specific data from said second plurality of data address locations in accordance with a specified universal service and protocol, and wherein said specified universal service and protocol is independent of the network-specific services and protocols of each of the networks.

17. The communication system of claim 16 wherein each of said first memories is an intermediate random access memory.

18. The communication system of claim 17 wherein each of said second memories is a flash read-only memory.

19. The communication system of claim 16 wherein each of said second plurality of data address locations has assigned an associated network-specific service and protocol.

WEST

Generate Collection Print

L1: Entry 4 of 6 File: PGPB Nov 21, 2002

DOCUMENT-IDENTIFIER: US 20020174178 A1
TITLE: Communication system for automation equipment based on the WSDL language

Abstract Paragraph (1):
Communication system on an IP network (50) between automation equipment (10) capable of executing at least one program (20) to provide automation functions and one or more remote items of equipment (30, 40) executing one or several computer applications. The communication system is conform with the WSDL (Web Services Description Language) language and provides monitoring, display, control, configuration or programming functions of the automation equipment (10) to remote equipment. The communication system uses at least one service description document (61) conform with the WSDL language accessible through a URL address and that describes capabilities of one or several WEB services (21, 21') capable of interacting with a program (20) on the automation equipment (10)

Summary of Invention Paragraph (1):
[0001] This invention relates to a communication system and a process for communication on a global Internet, Intranet or Extranet type network between at least one item of automation equipment offering one or several automation functions in an automation application and at least one item of remote equipment. The communication system uses the WSDL (Web Services Description Language), using at least one WEB service conform with the WSDL language and capable of interacting with a program in the automation equipment. This type of communication system may be applicable to any automation application particularly industrial logic controllers, building automation equipment, or instrumentation/control for electrical distribution networks.

Summary of Invention Paragraph (3):
[0003] It is known that this type of automation equipment may include a WEB server to be able to exchange data related to the automation equipment with a remote WEB client, such as a browser connected to a global communication network. This global network may be of the Internet, Intranet or Extranet type conform with the TCP/IP standard or the UDP/IP standard and will be called "IP network" throughout the rest of the presentation. In particular, these functions are described in documents WO9913418, U.S. Pat. No. 6,061,603, and U.S. Pat. No. 5,805,442. Data related to the automation equipment are then formatted and sent by the WEB server, for example in the HTML or XML (extended Markup Language) pages. It is also possible that a WEB server installed in automation equipment could load a program, usually called an Applet, in remote equipment, and the said program is executed in the remote equipment to exchange requests transported by the IP protocol, with the WEB server in the automation equipment.

Summary of Invention Paragraph (4):
[0004] In the following, the term "remote equipment" may denote a personal computer, a portable telephone, or a PDA (Personal Digital Assistant) type equipment, or a computer server such as an ASP (Applications Service Provider), an applications server (J2EE, NET, etc.), a WEB server, a WAP server, a database management system (DBMS) server, an integrated management software (PGI) server, an ERP (Enterprise Resource Planning) server, an EAI (Enterprise Application Integration) server, an electronic document management (EDM) server, a business to business electronic shopping (B-to-B) server, a station for programming automation equipment, or any other computer system. Remote equipment may also be used to refer to a set of remote items of equipment communicating with each other. The remote equipment comprises at

least one processing unit, is capable of connecting to at least one item of automation equipment through an IP network and executing a program or a set of computer programs. Some automation equipment such as dialog terminals may also be considered as remote equipment.

Summary of Invention Paragraph (5):
[0005] In the rest of this description, the term "intermediate equipment" refers to remote equipment connected to one or several remote items of equipment and to one or several items of automation equipment. The function of the intermediate equipment is to act as proxy for the automation equipment. It is also capable of receiving and sending requests encoded according to specific protocols on the IP network to automation equipment and it can interact with an automation equipment program. Some intermediate equipment such as network communication modules or network communication equipment may also be considered as being automation equipment.

Summary of Invention Paragraph (6):
[0006] The WSDL (Web Services Description Language) language is a language that can be used to make a simple description of WEB services in a distributed and decentralized environment. A WEB service is a modular application based on Internet that executes precise tasks and that respects a specific format. The WSDL language is based on the XML (extensible Markup Language) and may be used in combination with several other WEB protocols such as SOAP (Simple Object Access Protocol), HTTP (Hyper Text Transfer Protocol), HTTPS (Hyper Text Transfer Protocol/Secure Socket Layer), MIME (Multipurpose Internet Mail Extensions), SMTP (Simple Mail Transfer Protocol), FTP (File Transfer Protocol), or the IP (Internet Protocol) protocol. The WSDL language is based on XML schemes and provides a vocabulary defining a structure, contents and a communication description syntax. This language is deposited with the W3C (World Wide Web Consortium) and at the moment is described in a WSDL note 1.1 published on Mar. 15, 2001 and available from address http://www.w3.org/TR/wsdl.

Summary of Invention Paragraph (9):
[0009] It would be particularly useful for the designer of a computer application in remote equipment using development tools that are increasingly widespread on the market, or for a user of any computer application, for example such as a browser, a JAVA servlet, a Java Server Pages (JSP) application, an Active Server Pages (ASP) application, etc., running on an applications server or on a WEB server, to be able to make direct exchanges of data on an IP network with one or several items of automation equipment using the WSDL language. By adapting the automation equipment to WEB service description languages derived from the computer world, an automation equipment would thus be able to communicate with a remote computer application developed separately with development tools in the computer world, through WEB services, thus opening up automation applications to the Internet field. The WSDL language facilitates distribution of applications in the form of WEB services, particularly because the WSDL language offers an XML syntax that is capable of giving a WEB services description format.

Summary of Invention Paragraph (12):
[0012] In order to achieve this, the invention describes a communication system on an IP network between automation equipment with at least one processing unit capable of executing at least one program to provide automation functions and one or more remote items of equipment executing one or several computer applications. The communication system is conform with the WSDL (Web Services Description Language) language in order to provide automation equipment monitoring, display, control, configuration or programming functions to the remote equipment, and the communication system uses at least one service description document conform with the WSDL language that describes capabilities of one or several WEB services capable of interacting with an automation equipment program. A service description document is accessible for remote equipment using a URL, URI or IP address, through an IP network interface.

Summary of Invention Paragraph (13):
[0013] According to one characteristic, a WEB service is capable of receiving and sending messages encoded according to at least one communication protocol conform with at least one WSDL binding described in a service description document, on the

IP network.

Summary of Invention Paragraph (15):
[0015] According to another characteristic, the communication system comprises a service description document generator capable of dynamically building a service description document related to an automation equipment when requested by a remote equipment, and accessible to remote equipment through a URL, URI or IP address or through an IP network interface.

Summary of Invention Paragraph (16):
[0016] The invention also relates to a communication process used in such a communication system and comprising the following steps:

Summary of Invention Paragraph (18):
[0018] a second generation step in which a service description document conform with the WSDL language is used to generate all or part of a computer application by means of code generators or to generate a behaviour in a computer application by means of WSDL document interpreters, such that the computer application on remote equipment communicates with a WEB service through messages conform with the communication protocol described in the service description document,

Summary of Invention Paragraph (19):
[0019] a third communication step on the IP network between a computer application executed on remote equipment and at least one automation equipment WEB service, by means of messages conform with the communication protocol described in the service description document.

Brief Description of Drawings Paragraph (2):
[0021] FIG. 1 represents a first example of a communication system conform with the invention between automation equipment comprising a WEB service and a first remote equipment executing a computer application, which is partially or entirely generated by means of a development application executed in second remote equipment,

Brief Description of Drawings Paragraph (4):
[0023] FIG. 3 shows a variant to the communication process in FIG. 1 in which the service description document is memorized in a server,

Brief Description of Drawings Paragraph (5):
[0024] FIG. 4 represents a second example of a communication system conform with the invention between automation equipment comprising a WEB service and remote equipment executing a computer application that comprises a WSDL document interpreter,

Detail Description Paragraph (1):
[0028] With reference to FIG. 1, automation equipment 10 is connected through an IP network 50 to first remote equipment 30. The automation equipment 10 comprises at least one processing unit (not shown) capable of executing at least one program 20 offering one or several automation functions to an automation application. For example, this program 20 may be an application program (or a user program) for the instrumentation/control of an automation application, or it may also be the operating system installed in the automation equipment 10 directly. The automation equipment 10 comprises at least one WEB service 21 capable of interacting with the said program 20. In order to communicate on the IP network 50, the automation equipment 10 comprises an HTTP, HTTPS, SMTP, FTP, TCP, UDP or IP type network interface 15.

Detail Description Paragraph (2):
[0029] A WEB service is a resource accessible on an IP network through an IP network interface 15, 15', 15". According to the invention, a WEB service can receive and send messages 53 encoded according to a communication protocol conform with the WSDL binding described in a service description document 61, on the IP network 50. A WEB service is capable of interacting with a program 20 in automation equipment 10. This resource is formally described by a software interface contained in a service description document 61. An implementation of such a software interface is also called "WEB service" in this presentation.

Detail Description Paragraph (3):
[0030] The communication system described in this invention uses a service description document 61 that describes the capabilities of one or several WEB services 21, 21' on automation equipment 10, in other words that describes the WEB services that automation equipment 10 is capable of supplying or offering. The service description document 61 is accessible through a remote equipment 30, 40, either from its local resources or from remote resources identified by a Uniform Resource Locator (URL), Universal Resource Identifier (URI) or Internet Protocol (IP) address, through an IP network interface 15, 15', 15". Thus, due to this type of document 61, any remote equipment 30, 40 is able to know what services are available about an automation equipment 10 at all times. A service description document 61 may also contain a description of several standard WEB services 21, 21', for example corresponding to services systematically installed in a well-identified complete range of automation equipment. Similarly, it may also describe a particular WEB service 21, 21' available on a complete list of distinct automation equipment.

Detail Description Paragraph (9):
[0036] FIG. 2 diagrammatically shows a variant of FIG. 1 in which an intermediate equipment 70 is introduced and connected to the remote equipment 30, 40 and the automation equipment 10 through a network interface 151. The intermediate equipment 70, that may be a computer server, memorizes at least one service description document 61 and a WEB service 21' related to the automation equipment 10 and behaves like a Proxy that is a representative of the automation equipment 10. With this type of Proxy, remote equipment appears to be connected to the genuine automation equipment 10 although it is actually only connected to the proxy of the automation equipment 10. The WEB service 21' of the intermediate equipment 70 is capable of receiving and sending requests 54 encoded according to one or several protocols specific to the automation equipment, for example such as the PROFInet, EtherNet/IP, MODBUS/TCP, etc. protocols, on the IP network 50, to interact with at least one program 20 in automation equipment 10. According to one variant of the invention, the WEB service 21' of the intermediate equipment 70 is also capable of receiving and sending requests 54' encoded according to different protocols specific to the automation equipment, for example such as MODBUS, Uni-TE, AS-I, etc. protocols, without using an IP network, in order to interact with at least one program 20' in automation equipment 10'.

Detail Description Paragraph (10):
[0037] FIG. 3 shows another variant of the communication process in FIG. 1 according to which the automation equipment 10 installs a WEB service 21 but not a service description document 61. The service description document(s) 61 is (are) memorized in storage means 60" on a server 80. This server 80, that may be a file server or another server, is capable of communicating on the IP network 50 with remote equipment 30, 40 through a network interface 15", to supply the remote equipment with a service description document 61.

Detail Description Paragraph (11):
[0038] Thus, one of the advantages of this invention is also the fact that a service description document 61 describing the WEB services of automation equipment 10 may be built in different ways. It may be memorized either in storage means 60 located in the automation equipment 10 (FIG. 1), or in storage means 60' of an intermediate equipment 70 connected both to the automation equipment 10 and to the remote equipment 30, 40 (FIG. 2) or in local storage means (hard disk, CD-ROM, DVD or others) on remote equipment 40 communicating with the automation equipment 10, or remote storage means 60" in a server 80 accessible from the remote equipment 40 (FIG. 3) through an URL, URI or IP address through the IP network 50.

Detail Description Paragraph (15):
[0042] In FIG. 4, a second example shows a system for communication between automation equipment 10 comprising a WEB service 21 and a remote equipment 30 comprising a computer application 31, that uses the service description document 61 located in the automation equipment 10. Unlike the example in FIG. 1, the computer application 31 comprises a WSDL document interpreter 33. This WSDL document interpreter 33 reads the WSDL document by means of a read request 51 and will automatically generate a behaviour in the computer application 31, such as a man-machine interface (MMI) that corresponds to the operations defined in the WSDL

document. The interpreter 33 uses at least one WSDL binding conform with at least one of the protocols supported by the WSDL language such as SOAP, HTTP or MIME.

Detail Description Paragraph (17):
[0044] FIG. 5 uses the variant in FIG. 2 adapted to the example in FIG. 4. An intermediate equipment 70 connected to the remote equipment 30, 40 and to the automation equipment 10 through a network interface 15', memorizes at least one service description document 61 and a WEB service 21' related to the automation equipment 10.

Detail Description Paragraph (19):
[0046] FIG. 7 shows a variant of the example shown in FIG. 1 in which the same remote equipment 40 comprises a development application 41 and a computer application 31. The development application 41 is then capable of generating and locally deploying all or some of the computer application 31 to enable it to communicate with WEB services 21 of automation equipment 10. In this example, the development application 41 comprises a code generator 44 specific to the automation equipment. In this case, in an equivalent manner, it would also be possible to envisage that the development application 41 is a tool intended for the development of automation application programs conform with standard IEC 1131-3.

Detail Description Paragraph (20):
[0047] According to another embodiment of the invention, a service description document 61 may be generated dynamically at the time that a computer application 31 or a development application 41 of a remote equipment 30, 40, sends a request 51 to access services available in automation equipment 10. This dynamic generation function makes it possible to avoid systematically memorizing a service description document 61 and to generate it simply on request and therefore to be able to modify it for each request, particularly when the configuration of the automation equipment 10 is modified (for example following the connection or disconnection of an I/O module in a programmable logic controller 10). Furthermore, automation equipment 10 may want to show or conceal some WEB services depending on its state or depending on the remote equipment 30, 40 that wants to communicate with it.

Detail Description Paragraph (21):
[0048] In doing this, the communication system comprises a service description document generator 62 that is a program capable of dynamically creating a service description document 61. Preferably, this dynamic construction is achieved due to a service description document generator 62 when remote equipment 30, 40 requests access to the service description document 61 of the automation equipment 10. In this case, the service description document 61 does not need to be memorised since it is generated dynamically following each request 51 sent by a computer application to the service description document generator 62. However, in some cases, it would also be possible to envisage a dynamic document construction 61 initiated at the request of the automation equipment 10. Furthermore, it would also be possible to envisage a combined solution in which the generator 62 would be capable of dynamically personalizing a service description document 61 based on a standard document already memorized for an equipment family, and then by adapting its standard document dynamically as a function of the state of the automation equipment 10.

Detail Description Paragraph (30):
[0057] The invention also relates to a communication process used in a communication system like that described above. This process comprises the following steps:

Detail Description Paragraph (32):
[0059] A second generation step in which a service description document 61 conform with the WSDL language is used to automatically or manually generate 52 all or some of a computer application 31 (using code generators 43, 44) or to generate a behaviour in a computer application 31 (by means of WSDL document interpreters 33, 34) such that the computer application 31 on the remote equipment 30 communicates with a WEB service 21, 21' by means of messages 53 conform with the communication protocol described in the service description document 61.

Detail Description Paragraph (33):

[0060] A third communication step between a computer application 31 executed on remote equipment 30 and a WEB service 21, 21' on automation equipment 10 on the IP network 50 using messages 53 conform with the communication protocol described in the service description document 61.

Detail Description Paragraph (34):
[0061] During the generation step, a second remote equipment 40 executing a development application 41 can generate all or part of a computer application 31 either locally in the same second remote equipment 40 or remotely on the IP network 50 to a first remote equipment 30. Thus, the computer application 31 and the development application 41 may be on the same remote equipment 40 as shown in FIG. 7, or on two separate remote pieces of equipment communicating through the IP network 50 as shown in FIG. 1.

CLAIMS:

1. Communication system on an IP network (50) between an automation equipment (10) with at least one processing unit capable of executing at least one program (20) to provide automation functions and one or more remote items of equipment (30, 40) executing one or several computer applications, characterized by the fact that the communication system is conform with the WSDL (Web Services Description Language) language in order to provide monitoring, display, control, configuration or programming functions of the automation equipment (10) to the remote equipment (30, 40), and the communication system uses at least one service description document (61) conform with the WSDL language that describes capabilities of one or several WEB services (21, 21') capable of interacting with a program (20) on the automation equipment (10).

2. Communication system according to claim 1, characterized by the fact that a service description document (61) is accessible to remote equipment (30, 40) through a URL, URI or IP address through an IP network interface (15, 15', 15").

3. Communication system according to claim 2, characterized by the fact that a WEB service (21, 21') is capable of receiving and sending messages (53) encoded according to at least one communication protocol conform with at least one WSDL binding described in a service description document (61), on the IP network (50).

4. Communication system according to claim 3, characterized by the fact that at least one WSDL binding described in a service description document 61 is conform with the SOAP, HTTP or the MIME protocol.

5. Communication system according to claim 4, characterized by the fact that at least one service description document (61) describes the capacities of a WEB service (21, 21') to present a service conform with a protocol specific to the automation equipment.

6. Communication system according to claim 3, characterized by the fact that at least one WSDL binding described in a service description document (61) is conform with a protocol specific to the automation equipment.

7. Communication system according to claim 3, characterised by the fact that at least one WSDL binding described in a service description document (61) is conform with at least one version of the SOAP protocol encoded in a binary format.

8. Communication system according to claim 1, characterised by the fact that the service description document (61) for automation equipment (10) is memorized in local storage means of a remote equipment (30, 40).

9. Communication system according to claim 2, characterised by the fact that the service description document (61) related to automation equipment (10) is memorized in its storage means (60) located in the automation equipment (10).

10. Communication system according to claim 2, characterised by the fact that the service description document (61) related to automation equipment (10) is memorised in storage means (60') located in intermediate equipment (70) connected to the

automation equipment (10) and to at least one item of remote equipment (30).

11. Communication system according to claim 2, characterised by the fact that the service description document (61) for automation equipment (10) is memorized in remote storage means (60") located on a server (80).

12. Communication system according to claim 2, characterised by the fact that it comprises a service description document generator (62) capable of dynamically building a service description document (61) for automation equipment (10) following a request from a remote equipment (30, 40), and accessible to remote equipment (30, 40) through an URL, URI or IP address through an IP network interface (15, 15', 15").

13. Communication system according to claim 3, characterised by the fact that a WEB service (21) capable of interacting with a program (20) in automation equipment (10) is installed in the automation equipment (10).

14. Communication system according to claim 3, characterised by the fact that a WEB service (21') capable of interacting with a program (20) in automation equipment (10) is installed in intermediate equipment (70) connected to the automation equipment (10) and to at least one item of remote equipment (30, 40).

15. Communication system according to claim 14, characterised by the fact that the WEB service (21') is capable of receiving and sending requests (54, 54') encoded according to one or several protocols specific to the automation equipment.

16. Communication system according to claim 2, characterised by the fact that a discovery document for a service description document (61) related to automation equipment (10) is accessible to remote equipment (30, 40) through an URL, URL or IP address.

17. Communication system according to claim 16, characterised by the fact that the discovery document for a service description document (61) for automation equipment (10) is represented by one or several WEB pages conform with at least one WEB page description language, the said discovery document proposes one or several lists of URL, URL or IP addresses for one or several service description documents (61).

18. Communication system according to claim 16, characterised by the fact that the format of the discovery document of a service description document (61) related to automation equipment (10) is conform with the ADS (Advertisement and Discovery Services) specifications, or DISCO (Discovery) or UDDI (Universal Description, Discovery and Integration) specifications.

19. Communication system according to claim 16, characterised by the fact that the discovery document for a service description document (61) for automation equipment (10) is memorised in storage means (60) located in the automation equipment (10).

20. Communication system according to claim 16, characterised by the fact that the discovery document for a service description document (61) for automation equipment (10) is memorised in storage means (60') located in intermediate equipment (70) connected to the automation equipment (10) and at least one remote equipment (30).

21. Communication system according to claim 16, characterised by the fact that the discovery document for a service description document (61) for automation equipment (10) is memorised in remote storage means (60") located in a server (80).

22. Communication process used in a communication system according to one of the previous claims, characterized in that the process comprises the following steps: A first step for the discovery of a WEB service, in which a computer application (31) or a development application (41) executing in remote equipment (30, 40) sends a request (51) on the IP network (50) to receive one or more service description documents (61) conform with the WSDL language and describing the capabilities of one or several WEB services (21, 21'), A second generation step in which a service description document 61 conform with the WSDL language is used to automatically or manually generate (52) all or some of a computer application (31) using code

generators (43, 44,) or to generate a behaviour in a computer application (31) (by means of WSDL document interpreters (33, 34) such that the computer application (31) on the remote equipment (30) communicates with a WEB service (21, 21') by means of messages (53) conform with the communication protocol described in the service description document (61). A third communication step between a computer application (31) executed on remote equipment (30) and a WEB service (21, 21') on automation equipment (10) on the IP network (50) using messages (53) conform with the communication protocol described in the service description document (61).

23. Communication process according to claim 22, characterised by the fact that during the discovery step, a computer application (31) or a development application (41) may use one or several discovery documents to discover one or several service description documents (61) describing a WEB service (21, 21') related to automation equipment (10).

24. Communication process according to claim 22, characterised by the fact that during the generation step, a development application (41) being executed in a second remote equipment (40) may generate all or some of a computer application (31) locally in the second remote equipment (40) or in a first remote equipment (30) through the IP network (50).

WEST

Generate Collection | Print

L1: Entry 5 of 6 File: PGPB Sep 12, 2002

DOCUMENT-IDENTIFIER: US 20020129333 A1
TITLE: System and method for programmatically generating a graphical program based on a sequence of motion control, machine vision, and data acquisition (DAQ) operations

Summary of Invention Paragraph (4):
[0004] Computer-based motion control involves precisely controlling the movement of a device or system. Computer-based motion control is widely used in many different types of applications, including applications in the fields of industrial automation, process control, test and measurement automation, robotics, and integrated machine vision, among others. A typical computer-based motion system includes components such as the moving mechanical device(s), a motor with feedback and motion I/O, a motor drive unit, a motion controller, and software to interact with the motion controller.

Detail Description Paragraph (48):
[0077] FIG. 2B illustrates an exemplary industrial automation system 160 which may implement embodiments of the invention. The industrial automation system 160 is similar to the instrumentation or test and measurement system 100 shown in FIG. 2A. Elements which are similar or identical to elements in FIG. 2A have the same reference numerals for convenience. The system 160 includes a host computer 82 that comprises a CPU, a display screen, memory, and one or more input devices such as a mouse or keyboard as shown. The host computer 82 may connect to one or more devices or instruments to interact with a process or device 150 to perform an automation function, such as MMI (Man Machine Interface), SCADA (Supervisory Control and Data Acquisition), portable or distributed data acquisition, process control, advanced analysis, or other control.

Detail Description Paragraph (53):
[0082] The DAQ card 114, the PXI chassis 118, the video device 132, and the image acquisition card 134 may be connected to the computer 82 as described above. The serial instrument 182 may be coupled to the computer 82 through a serial interface card 184, or through a serial port, such as an RS-232 port, provided by the computer 82. The PLC 176 may couple to the computer 82 through a serial port, Ethernet port, or a proprietary interface. The fieldbus interface card 172 may be comprised in the computer 82 and may interface through a fieldbus network to one or more fieldbus devices. Each of the DAQ card 114, the serial card 184, the fieldbus card 172, the image acquisition card 134, and the motion control card 138 are typically plugged in to an I/O slot in the computer 82 as described above. However, these cards 114, 184, 172, 134, and 138 are shown external to computer 82 for illustrative purposes. In typical industrial automation systems a device will not be present of each interface type, and in fact many systems may only have one or more devices of a single interface type, such as only PLCS. The devices may be coupled to the device or process 150.

Detail Description Paragraph (87):
[0116] In one embodiment, the prototyping environment may interact with another application to programmatically generate the graphical program, such as a graphical programming development environment application. For example, the graphical programming development environment may provide an application programming interface (API) for programmatically generating graphical programs.

Detail Description Paragraph (206):

[0235] Through the API 504, the client program 502 may communicate with the server program 506. The server program 506 is operable to perform the actions indicated by the API calls. For example, the server program may be operable to create a new graphical program, add objects to the graphical program, connect graphical program objects, etc. The API calls may also enable the client program 502 to request an existing graphical program to be modified. Thus, in one embodiment, in response to the user editing an existing motion control sequence, a graphical program corresponding to the motion control sequence may be programmatically modified to reflect the changes.

Detail Description Paragraph (209):
[0238] The LabVIEW environment provides specialized support for developers of instrumentation and industrial automation applications, and a LabVIEW graphical program may be referred to as a "virtual instrument" or "VI". The LabVIEW environment comprises functionality referred to as "VI Server" which enables client programs to communicate with the LabVIEW environment. The VI Server functionality enables client programs to create or edit a LabVIEW graphical program or VI.

Detail Description Paragraph (224):
[0253] After a graphical program to implement a motion control sequence has been generated, the user may desire to modify the motion control sequence. The generated graphical program may then be updated in response. In one embodiment, the existing graphical program may simply be discarded, and a new graphical program may be generated based on the updated sequence. However, in another embodiment, the existing graphical program may be programmatically modified to reflect the change made to the motion control sequence. For example, if the user has made changes or additions to the graphical program, it may be advantageous to programmatically modify the affected portion of the graphical program, preserving the user's changes. Also, the ability to programmatically modify the graphical program may be useful for interactive stepwise creation of the graphical program, such as described above with reference to FIG. 9.

WEST

End of Result Set

Generate Collection Print

L1: Entry 6 of 6 File: PGPB Sep 12, 2002

DOCUMENT-IDENTIFIER: US 20020126151 A1
TITLE: System and method for graphically creating a sequence of motion control, machine vision, and data acquisition (DAQ) operations

Summary of Invention Paragraph (4):
[0003] Computer-based motion control involves precisely controlling the movement of a device or system. Computer-based motion control is widely used in many different types of applications, including applications in the fields of industrial automation, process control, test and measurement automation, robotics, and integrated machine vision, among others. A typical computer-based motion system includes components such as the moving mechanical device(s), a motor with feedback and motion I/O, a motor drive unit, a motion controller, and software to interact with the motion controller.

Summary of Invention Paragraph (9):
[0008] Motion control applications often involve other types of instrumentation technologies in addition to motion control, e.g., to implement a test and measurement, industrial automation, process control, or other type of application. In particular, applications that utilize motion control often incorporate data acquisition (DAQ) functionality as well, e.g., to acquire and analyze data. As one example, DAQ functionality may be used to acquire test data for a product on an assembly line. The test data may then be analyzed for quality control purposes, and motion control functionality may be used to place the product in an appropriate bin based on the results of the analysis.

Detail Description Paragraph (48):
[0093] FIG. 2B illustrates an exemplary industrial automation system 160 which may implement embodiments of the invention. The industrial automation system 160 is similar to the instrumentation or test and measurement system 100 shown in FIG. 2A. Elements which are similar or identical to elements in FIG. 2A have the same reference numerals for convenience. The system 160 includes a host computer 82 that comprises a CPU, a display screen, memory, and one or more input devices such as a mouse or keyboard as shown. The host computer 82 may connect to one or more devices or instruments to interact with a process or device 150 to perform an automation function, such as MMI (Man Machine Interface), SCADA (Supervisory Control and Data Acquisition), portable or distributed data acquisition, process control, advanced analysis, or other control.

Detail Description Paragraph (53):
[0098] The DAQ card 114, the PXI chassis 118, the video device 132, and the image acquisition card 134 may be connected to the computer 82 as described above. The serial instrument 182 may be coupled to the computer 82 through a serial interface card 184, or through a serial port, such as an RS232 port, provided by the computer 82. The PLC 176 may couple to the computer 82 through a serial port, Ethernet port, or a proprietary interface. The fieldbus interface card 172 may be comprised in the computer 82 and may interface through a fieldbus network to one or more fieldbus devices. Each of the DAQ card 114, the serial card 184, the fieldbus card 172, the image acquisition card 134, and the motion control card 138 are typically plugged in to an I/O slot in the computer 82 as described above. However, these cards 114, 184, 172, 134, and 138 are shown external to computer 82 for illustrative purposes. In typical industrial automation systems a device will not be present of each interface

type, and in fact many systems may only have one or more devices of a single interface type, such as only PLCs. The devices may be coupled to the device or process 150.

Detail Description Paragraph (87):
[0130] In one embodiment, the prototyping environment may interact with another application to programmatically generate the graphical program, such as a graphical programming development environment application. For example, the graphical programming development environment may provide an application programming interface (API) for programmatically generating graphical programs.

Detail Description Paragraph (206):
[0233] Through the API 504, the client program 502 may communicate with the server program 506. The server program 506 is operable to perform the actions indicated by the API calls. For example, the server program may be operable to create a new graphical program, add objects to the graphical program, connect graphical program objects, etc. The API calls may also enable the client program 502 to request an existing graphical program to be modified. Thus, in one embodiment, in response to the user editing an existing motion control sequence, a graphical program corresponding to the motion control sequence may be programmatically modified to reflect the changes.

Detail Description Paragraph (209):
[0235] The LabVIEW environment provides specialized support for developers of instrumentation and industrial automation applications, and a LabVIEW graphical program may be referred to as a "virtual instrument" or "VI". The LabVIEW environment comprises functionality referred to as "VI Server" which enables client programs to communicate with the LabVIEW environment. The VI Server functionality enables client programs to create or edit a LabVIEW graphical program or VI.

Detail Description Paragraph (224):
[0247] After a graphical program to implement a motion control sequence has been generated, the user may desire to modify the motion control sequence. The generated graphical program may then be updated in response. In one embodiment, the existing graphical program may simply be discarded, and a new graphical program may be generated based on the updated sequence. However, in another embodiment, the existing graphical program may be programmatically modified to reflect the change made to the motion control sequence. For example, if the user has made changes or additions to the graphical program, it may be advantageous to programmatically modify the affected portion of the graphical program, preserving the user's changes. Also, the ability to programmatically modify the graphical program may be useful for interactive stepwise creation of the graphical program, such as described above with reference to FIG. 9.

WEST

Generate Collection Print

L2: Entry 22 of 40 File: USPT Aug 15, 2000

DOCUMENT-IDENTIFIER: US 6105093 A
TITLE: Interface monitor for communicating between different communication protocols

Abstract Text (1):
An interface monitor is connected with a LAN that is formatted with an RS-485 protocol and which is interconnected with a plurality of electrical distribution devices. The monitor includes a socket connection for receiving an RS-232 plug from a PC and software logic for interpreting between the RS-232 and RS-485 formats. The PC is enabled for communicating with each one of the electrical distribution devices.

Brief Summary Text (6):
U.S. Pat. No. 5,127,090 entitled "Map Interface Unit (MAP) for Industrial Programmable Logic Controllers" describes a manufacturing automation protocol (MAP) interface unit that couples a LAN, connecting with programmable logic controllers, to a MAP network.

Brief Summary Text (7):
U.S. Pat. No. 5,245,703 entitled "Data Processing System With Multiple Communication Buses and Protocols" discloses an interface unit for connecting between internal and external communication buses having different protocols.

Brief Summary Text (8):
U.S. Pat. No. 5,355,365 entitled "Intelligent Local Area Network Modem Mode" describes a LAN modem node that allows PC's on the LAN to communicate with off-LAN PC's, printers and other local area networks.

Brief Summary Text (9):
When there is a need for a host PC to communicate with any of the electrical equipment on the LAN, as described within the aforementioned U.S. patent application Ser. No. 08/628,533, it would be convenient to interpose a monitor unit having capacity to receive both RS-232 configured and RS-485 configured plug sockets along with the capacity to translate between the associated RS-232 and RS-485 protocols.

Brief Summary Text (12):
An interface monitor is connected with a LAN that is formatted with an RS-485 protocol and which is interconnected with a plurality of electrical devices. The monitor includes a socket connection for receiving an RS-232 plug from a PC and includes sufficient software logic for interpreting between the RS-232 and RS-485 formats. The PC is enabled for communicating with each one of the electrical devices.

Detailed Description Text (2):
As shown in FIG. 1, a number of distributed electrical metering and control devices 11-14, such as relays, meters, circuit breakers and the like described in the aforementioned U.S. patent application Ser. No. 08/628,533, are interconnected by means of a cable connector 15 with a LAN 10 containing RS-485 protocol. A good example of a circuit breaker having such communication facility is found in U.S. Pat. No. 4,675,481 entitled "Circuit Breaker and Protective Relay Unit". When a PC 18 such as an IBM 350-P75 is required to address any of the devices to request status and other information, the PC is connected by means of a cable connector 19 with a monitor unit 17 in the form of an enclosure that contains a plurality of circuit components which will be described later with reference to FIG. 3. The

monitor 17 connects with the LAN by means of a cable connector 16. Since the cable connector 19 is removably-connectable with the monitor, various PC's can be connected at different times to access the LAN without requiring any modification to the PC or the monitor to make the connection.

Detailed Description Text (3):
This feature is best understood by now referring to FIG. 2 wherein the PC 18 is shown having an RS-232 socket or port 20 arranged on a rear surface thereof. The enclosure 9 that defines the monitor 17 includes both an RS-232 port 30 for receiving the cable connector 19 from the PC 18 as well as an RS-485 port 31 for receiving the cable 32 that connects with the LAN 10. The electrical device 11, consisting of a circuit breaker, meter or the like, connects with the LAN 10 by means of the cable 15 and the RS-485 port 33. Communication between the RS-232 port 30 and the RS-485 port 31 is made by a conversion algorithm resident within the monitor microprocessor described earlier. It is thus seen that at different times, different PC's 18 can communicate with any number of devices 11 by the use of a single monitor 17, in accordance with the teachings of the invention.

Detailed Description Text (4):
As shown in FIG. 3, the monitor 17 includes a CPU in the form of a microprocessor 35 connecting with a display driver 46 over rail 42 and with a keyboard controller 47 over rail 40 to provide external keypad and display facility. The microprocessor connects with a RAM 36 over rail 44 and the RAM contains a program for converting RS-232 protocol resident in the PC modem with the protocol employed within the LAN 10 of FIG. 1 in the manner to be described below. To provide timing facility, the microprocessor connects with a clock 37 over rail 45. In further accordance with the teachings of the invention, the RS-232 port 30 includes a Universal Address Radio Transceiver "UART" 39 such as an SP type 16C550 configured for RS-232 communications and connects with the microprocessor over rail 41. The RS-485 port 31 includes a similar UART 38 configured for RS-485 communication. As shown earlier, in FIG. 2, the PC 18 connects with the RS-232 port 20 and the LAN 10 connects with the RS-485 port 31. The conversion program within the RAM 36 operates in the following manner. When a message is received on the RS-232 port 30, the UART 39 passes a byte to the microprocessor port 1 over rail 41. The microprocessor 35 then copies the message to port 2, which transmits the message to the RS-485 port 31 over rail 43. When a message is received at the RS-485 port 31, the UART 38 passes a byte to the microprocessor port 2 over rail 43. The microprocessor 35 then copies the message to port 1, which transmits the message to the RS-232 port 30 over rail 41.

Detailed Description Text (5):
To provide address between any of the electrical devices 11-14 of FIG. 1, the Flow Chart 21 depicted in FIG. 4 is employed as follows. Auto detection (22) provides a means of self-configuring by scanning the LAN for all devices 11-14. The auto detection program resident in the monitor 17 (FIG. 2) scans all device addresses and waits for a response from the device. By inspecting the response received, the device characteristics can be determined. Absence of a device at a particular address is determined when a time-out condition occurs. Two retries are attempted to determine if a device is at a given address.

Detailed Description Text (7):
The main form is displayed (23) on the PC screen upon startup of the Autodetection program allowing the PC operator to initialize the status structure (24) and the internal PC transmitter and receiver buffers (25).

Detailed Description Text (9):
Another application of the monitor 17 shown in FIGS. 1 and 2 is to provide standby master control facility to the devices listed within the aforementioned U.S. patent application Ser. No. 08/628,533, in the event the master server becomes disabled. The monitor having acquired the function and location of all the devices within the LAN is then able to communicate with the devices to provide supervisory data acquisition and control function.

CLAIMS:

1. A communications monitor unit for intercommunication between an external computer

and a network including a plurality of electrical metering and control devices; the monitor unit comprising:

an enclosure;

a processor disposed within the enclosure;

a first port coupled to the external computer for transmitting data between the external computer and the communications monitor in an RS-232 communications protocol;

a second port coupled to the network for transmitting the data between the network and the communications monitor in an RS-485 communications protocol; and

an auto detection program resident in the communications monitor;

wherein said processor executes said auto detection program to determine a characteristic of each of the plurality of electrical metering and control devices.

5. The monitor of claim 1 wherein said processor executes said auto detection program to scan addresses of the plurality of electrical metering and control devices.

7. The monitor of claim 1 wherein said processor executes said auto detection program to populate a database file for storing information about the plurality of electrical metering and control devices.

8. A power management system for monitoring a plurality of electrical metering and control devices; said system comprising:

a network interconnecting the plurality of electrical metering and control devices;

an external computer for monitoring the plurality of electrical metering and control devices;

a communications monitor unit including:

a processor;

a first port coupled to the computer for transmitting data between the external computer and the communications monitor unit in an RS-232 communications protocol;

a second port coupled to the network for transmitting data between the plurality of electrical metering and control devices and the communications monitor unit in an RS-485 communications protocol;

an auto detection program resident in the communications monitor unit; and

a database file for storing information about the plurality of electrical metering devices;

wherein the processor executes said auto detection program to populate said database file.

11. The management system of claim 8 wherein said processor executes said auto detection program to determine a characteristic of each of the plurality of electrical metering and control devices.

12. The management system of claim 8 wherein said processor executes said auto detection program to scan addresses of the plurality of electrical metering and control devices.

14. The monitor of claim 8 wherein said communications monitor communicates with the plurality of electrical metering and control devices and provides supervisory control to the plurality of electrical metering and control devices upon failure of

the external computer.

15. A method for requesting and receiving information from a plurality of electrical metering and control devices located on a communications network, the communications network including an external computer arranged to communicate with a monitor computer in the RS-232 protocol, the monitor computer arranged to communicate with the plurality of electrical metering and control devices in the RS-485 protocol, the method comprising:

(a) sending messages from the monitor computer to each electrical metering and control device at a predetermined baud rate using the RS-485 protocol;

(b) waiting a predetermined time period for a response from each device;

(c) for each device that does not respond, lowering the baud rate by a predetermined increment and repeating steps (a) and (b); and,

(d) transmitting the response from the monitor computer to the external computer in an RS-232 protocol.

WEST

Generate Collection Print

L2: Entry 24 of 40 File: USPT Nov 2, 1999

DOCUMENT-IDENTIFIER: US 5978578 A
TITLE: Openbus system for control automation networks

Abstract Text (1):
A novel control automation system for enabling I/O boards to access communication networks for receiving and transmitting real time control information over a communication network is disclosed. The system includes a control bus, a node controller and a development system. External hardware that connects to I/O devices such as sensors, motors, monitors, machines, etc. can be connected to the invention via I/O boards that receives and transmit digital signals, representing control information, to the bus. The bus functions as the hub of operation, receiving network communications, processing cooperative logic and transmitting information over the communication network. The bus enables single or multiple controllers to access real time information generated by the attached hardware. The bus also enables the execution of I/O operations that originated in external controllers and transmitted over the communication network. The bus allows any I/O control board having a common interface, such as ISA, PCI, Compact PCI, etc., to connect to the bus by attachment to one of its slots. An intelligent embedded implementation process provides the logic necessary to enable the connectivity between the I/O boards and the communication network. The development system includes a real-time compiler for generating p-code to be executed on the real-time kernel running in the node controller. The real-time compiler generates p-code from the combination of event triggers, event actions and program logic making up the user's application.

Brief Summary Text (2):
The present invention relates generally to computer communication networks and more particularly relates to a system for implementing a control automation network.

Brief Summary Text (6):
The expected benefits of having open and modular architecture controllers include reduced initial investments, low life cycle costs, maximized machine uptime, minimized machine downtime easy maintenance of machines and controllers, easy integration of commercial and user proprietary technologies, plug and play of various hardware and software components, efficient reconfiguration of controllers to support new processes, incorporation of new technologies and the integration of low cost, high speed communication in machining lines for transferring large amounts of data.

Brief Summary Text (8):
In today's large automation market, there is a growing number of PC board manufacturers that produce a variety of boards. These boards are targeted towards automation implementation that use the PC and the control platform. Since automation data networks implements a proprietary technology that are not very open for `Intranet communication.`

Brief Summary Text (10):
A high level block diagram illustrating an example prior art proprietary control network including proprietary programmable logic controllers, sensors and I/O devices is shown in FIG. 1. A proprietary network 33 (e.g., Fieldbus) forms the core of the automation control system. Connected to this network are programmable logic controllers (PLCs) 34 which as also proprietary. Connected to the PLCs 34 are the sensors and other I/O devices 32. The proprietary PLCs implement the Automation and Control Layer functionality and the sensors and I/O devices implement the

Information and Device Layer.

Brief Summary Text (16):
It is another object of the present invention to provide an automated control system that provides automated control over standard communication networks.

Brief Summary Text (17):
Another object of the present invention is to provide an automated control system that permits the communication of data between various sensors and I/O devices and conventional networks.

Brief Summary Text (20):
The present invention comprises a novel control automation system for enabling I/O boards to access communication networks for receiving and transmitting real time control information over a communication network. The system includes a control bus, a node controller and a development system. External hardware that connects to I/O devices such as sensors, motors, monitors, machines, etc. can be connected to the invention via I/O boards that receives and transmit digital signals, representing control information, to the bus. The bus functions as the hub of operation, receiving network communications, processing cooperative logic and transmitting information over the communication network. The bus enables single or multiple controllers to access real time information generated by the attached hardware. The bus also enables the execution of I/O operations that originated in external controllers and transmitted over the communication network. The bus allows any I/O control board having a common interface, such as ISA, PCI, Compact PCI, etc., to connect to the bus by attachment to one of its slots. An intelligent embedded implementation process provides the logic necessary to enable the connectivity between the I/O boards and the communication network.

Brief Summary Text (21):
The development system includes a real-time compiler for generating p-code to be executed on the target system. The target system, e.g., the node controller, runs the real-time kernel. The target system can be a PC running a commercially available operating system such as Windows NT, VxWorks, Lynx, etc. The real-time compiler generates p-code from the combination of event triggers, event actions and program logic making up the user's application. External input signals and entities such as variables, timers, etc. are analyzed and used to trigger events in the real-time kernel. Based on the program logic as expressed in the p-code, various actions are taken in response to changes in the values of the external input signals and/or entities. The real-time kernel functions to implement a state machine that receives inputs and generates outputs. The actions taken by the system are represented as a sequence of frames with each frame representing a unit of action.

Brief Summary Text (23):
There is therefore provided in accordance with the present invention a control automation system for controlling a plurality of input and output (I/O) devices in accordance with a user's application, the system connected to a network for communicating control automation information, the system comprising a development system optionally coupled to the network, the development system generating p-code embodying event triggers, event actions and program logic implementing the user's application, and at least one node controller coupled to the network for executing in real-time the p-code generated by the development system.

Brief Summary Text (25):
The development system comprises a real-time compiler for generating p-code in accordance with the event triggers, event actions and program logic of the user's application. The kernel means comprises an external input signal scanner for reading, storing and determining changes to external input signals received from the plurality of I/O devices, an event triggers evaluation module for detecting changes to the external input signals and internal entities, the event triggers evaluation module for determining and resolving all event triggers corresponding to the detected changes, a scheduler for marking all actions corresponding to the event triggers that resolve true, an action execution unit for executing and implementing the actions marked for execution by the scheduler, and an entity processor for determining any changes to values assigned to an entity, the entity processor

notifying the event triggers evaluation module of the entity value changes.

Brief Summary Text (27):
There is also provided in accordance with the present invention a node controller apparatus for use in a control automation system, the system for controlling a plurality of input and output (I/O) devices in accordance with a user's application, the system including a network for communicating control automation information, the apparatus comprising processor means for managing and controlling the operation of the node controller, the processor means for executing a real-time kernel, the kernel implementing the user's application embodied in p-code, network interface means for connecting the node controller to the network, I/O interface means for connecting the node controller to the plurality of I/O devices, and bus means for interconnecting together the processor means, the kernel means, the network interface means and the I/O interface means.

Brief Summary Text (31):
In addition, there is provided in accordance with the present invention, in a computer system, a method of generating p-code for execution on a node controller as part of a control automation system for controlling a plurality of input and output (I/O) devices in accordance with a user's application, the application including event triggers, event actions and program logic, the method comprising the steps of generating a plurality of pointer tables, each pointer table associated with either an external input signal or an entity, each pointer table comprising a plurality of pointer entries, each pointer entry pointing to an event trigger, generating an event trigger table, the event trigger table comprising a plurality of event trigger entries, each event trigger entry corresponding to an action that references the particular external input signal or entity that points thereto, generating a plurality of actions, each of the actions comprising at least one frame, the actions, the actions representing the generation of output signals and/or the modification of the internal entities, and wherein the plurality of pointer tables, the event trigger table and the plurality of actions generated in accordance with the event triggers, event actions and program logic making up the user's application.

Brief Summary Text (32):
There is also provided in accordance with the present invention a node controller apparatus for use in a control automation system, the system for controlling a plurality of input and output (I/O) devices, the system including a network for communicating control automation information, the apparatus comprising processor means for managing and controlling the operation of the node controller, network interface means for connecting the node controller to the network, I/O interface means for connecting the node controller to the plurality of I/O devices, and bus means for interconnecting together the processor means, the network interface means and the I/O interface means.

Brief Summary Text (33):
Further, there is provided in accordance with the present invention a node controller apparatus for use in a control automation system, the system for controlling a plurality of input and output (I/O) devices in accordance with a user's application, the system including a network for communicating control automation information, the apparatus comprising processor means for managing and controlling the operation of the node controller, the processor means for executing a real-time kernel, the kernel implementing the user's application embodied in p-code, network interface means for connecting the node controller to the network, and bus means for interconnecting together the processor means, the kernel means and the network interface means.

Drawing Description Text (3):
FIG. 1 is a high level block diagram illustrating an example prior art proprietary control network including proprietary programmable logic controllers, sensors and I/O devices;

Detailed Description Text (2):
The present invention is a system for providing computer operated real-time process control with the means for interacting with an external system. The present

invention also provides a development system comprising a computer compiler for generating real-time code executable on a real-time kernel that resides in a target system. In addition, the present invention provides automation control over standard communication networks such as Ethernet and ATM. The system comprises an intelligent network I/O node controller for automation control that has a common interface with external processors and compilers. In addition, the network I/O node controller implements local logic to create an intelligent controller. A key aspect of the present invention is that automation control information can be transmitted on a conventional backbone network using a conventional connectivity protocol without the need for dual networks, i.e., one for standard data and one for automation control information. Further, a control bus in the intelligent network I/O node controller permits the use of off the shelf I/O cards for interfacing the node controller to input and output devices.

Detailed Description Text (3):
As stated previously, since conventional automation networks implement proprietary technologies they are not well suited for open Intranet communications. The OpenBus system of the present invention functions to fill the void to provide the infrastructure or `infranet` for communications within a control environment. Sensor and actuator level data is managed locally within the infranet but can be shared with higher level data networks through Intranets or other networking platforms. Using open APIs, devices within the infranet share process data and device status information with other nodes via the Intranet or the Internet. The OpenBus system of the present invention enables communication from sensors and actuators on the plant floor to the plant manager's desk anywhere in the world via the Internet, for example, resulting in a seamless network from I/O to the Internet.

Detailed Description Text (4):
OpenBus connectivity can be combined with Java applets in industrial applications making it possible for a plant manager, for example, to monitor, change or control any element of the industrial control system from the sensor all the way to a high level information system. Plant maintenance personnel can access devices at any point in the network, gather data and make modifications. Service technicians can download new software to devices in the field using Java applets received through an Intranet or Internet connection. If technical support is required, a direct line can be established with a customer support representative to diagnose and repair devices remotely.

Detailed Description Text (8):
Openness is required here because no one vendor offers the entire scope of host computers, software and communication interfaces, such as computer cards, bridges, routers and media. Considering industrial automation, Ethernet, primarily TCP/IP, has become a de facto standard for the Information Layer. Users purchase products from multiple vendors expecting openness. Control vendors therefor support Ethernet in their controllers, supervisory software and drivers.

Detailed Description Text (9):
The Automation and Control Layer 62 comprises DCS controllers, programmable logic controllers (PLCs), I/O chassis, dedicated human interfaces, motor drives and PCs. This layer is the core of the architecture that bridges the Information and Device Layers, enabling communication throughout the enterprise. Responses here must be in the order of milliseconds to be considered real time. For the Automation and Control Layer the driving force is the need for deterministic data delivery between controllers and I/O devices.

Detailed Description Text (12):
Devices are less complex at the Device Layer than they are at the Information Layer but they are more diverse. The size and cost to imbed connections in a device are critical at this layer. For example, consider adding a network connection to a $ 70 photoeye. In addition, no single vendor can offer all the possible devices, e.g., sensors and actuators, a user could need. For a true device network, the actual devices must be interoperable from manufacture to manufacturer. An I/O device can be taken from one network and be replaced with an I/O device from another network while the operation of the system behaves the same. The present invention provides this level of interoperability by using standard communication control networks such as

Ethernet or FDDI on the one hand and by permitting third party connectivity via standard PC I/O boards using standard PC buses such as PCI, EISA or VME.

Detailed Description Text (18):
A high level block diagram illustrating an automation control network constructed in accordance with an embodiment of the present invention is shown in FIG. 3. At the core of the system, termed OpenBus, is the OpenBus node controller 10. Each OpenBus node controller is connected to the network 18 which may be, for example, a LAN, WAN, the Internet, Intranet or any other suitable data, control or area network. Personal computers 14 are also connected to the network. The PCs function to execute various application programs constituting the Information Layer. In addition, the p-code control application can be executed on PCs as well. A gateway 42 provides connectivity to external networks such as the Internet 40.

Detailed Description Text (21):
As stated previously, openness in an industrial control automation network is very desirable. More specifically, within the network, openness is very important at the Control Layer. In the past, this layer was designed by committees which is difficult resulting in a lack of sufficient openness. The system of the present invention functions to open the Device Layer, e.g., sensors, starters, I/O blocks, etc., and to allow third parties to supply I/O control devices that would be able to connect to the network via the OpenBus system using PC boards built according to standards such as PCI, ISA, EISA, VME, etc.

Detailed Description Text (22):
Conventional area networks such as Ethernet, ATM, FDDI, etc., under the present invention, comprise independent intelligent network nodes. Each OpenBus intelligent node comprises an imbedded processor that functions to intermediate between the I/O boards and the area network. A set of high level API's can be written that allow each processor, controller or computer connected to the area network or the Internet to access sensor information at the application connectivity layer. In addition, the intelligent nodes participate in a distributed processing control environment by implementing independent local functionality that was previously programmed using the development system.

Detailed Description Text (24):
An embedded processor 22 controls and manages the node controller, functioning to control the communication between the interface circuitry boards and the NIC. The embedded processor is capable of executing Java applets 50 and application p-code control applications 52 developed on the development system. The local bus permits certain portions of an application program to be implemented in the node controller as a form of distributed or cooperative automated control processing. Further, the NIC and the I/O boards permit the local attachment of various analog and digital sensors, thus creating an integrated smart sensor attached to the network.

Detailed Description Text (26):
The OpenBus system of FIGS. 3 and 4 enables I/O boards to access conventional communication networks for the receipt and transmission of real-time control information over the network. Hardware that includes I/O attachment (e.g., sensors, motors, monitors, machines, etc.) are connected to the OpenBus via I/O boards that receive/transmit signals representing control information over the bus.

Detailed Description Text (27):
The OpenBus node controller 10 functions as the hub of operation of the system. It receives network communications, processes cooperative logic and transmits information over the network. The system permits multiple controllers, such as PCs 14 or node controllers 10, to access real time information generated by attached hardware 26 located anywhere in the network. A node controller 10 can execute I/O operations that originated in PCs 14 or other controllers 10 and transmitted over the network 18 and/or the Internet 40 communication networks.

Detailed Description Text (32):
A high level logic flow diagram illustrating the embedded open bus control process of the present invention is shown in FIG. 5. With reference to FIGS. 3 and 4, the first action performed by the embedded processor 22 upon power up is to initialize

the node controller (step 70). Once initialized, the processor loads the embedded program into memory (step 72). The embedded program comprises control programs developed by the development system written and compiled into p-code. In addition, the development system can generate Java scripts or applets. Then, the processor goes out on the bus and identifies each of the I/O boards 20 installed on the bus (step 74). Once the I/O boards are identified, the processor attempts to establish communications with the attached network 18 via NIC 24 (step 76).

Detailed Description Text (34):
The first step of the network communication management process is to wait for a network communication (step 78). Once a network communication is received, the processor checks if it is a network message (step 80). If it is not a network message control returns to step 78. If it is a network message, the message is then analyzed (step 82) and the dispatcher is activated (step 84). Note that optionally, the network communication management process can be implemented in Java code. The dispatcher is described in more detail below.

Detailed Description Text (41):
The level of openness required for an application varies with and is dependent upon the functionality of the communication layer and the types of devices found at that layer. Openness is usually achieved by the use of standards. These standards are either sanctioned by an official body, e.g., IEC/ISA SP50 Fieldbus, or is commonly accepted enough to become a de facto standard, e.g., Ethernet TCP/IP. Many vendors and end users prefer de facto standards over official standards because they result in a shorter time to market and have a singular customer and application focus.

Detailed Description Text (47):
The Filbus is based on distributed intelligence and peer to peer communication. Firmware functions are built into each Filbus I/O module and enable basic capabilities such as pulse count, delay before action and sending/receiving messages to/from other modules on the network. The Filbus runs at 375 Kbps, permits a maximum of 250 nodes, uses master/slave arbitration, uses twisted pair cable and has application in data acquisition.

Detailed Description Text (49):
The Bitbus was originally introduced by Intel Corporation as a way to add remote I/O capability to Multibus systems. This original Fieldbus is one of the most mature and most broadly used networks today. Bitbus permits programs to be downloaded and executed in a remote node providing for distributed system configuration. The Bitbus runs at 375 Kbps, permits a maximum of 250 nodes, uses master/slave arbitration, uses twisted pair cable and has application in process control.

Detailed Description Text (51):
The Worldfip provides a deterministic scheme for communicating process variables. Worldfip uses an original mechanism whereby the bus arbitrator broadcasts a variable identifier to all nodes on the network, triggering the node producing that variable to place its value into the network. This feature eliminates the notion of node address and makes it possible to design distributed process control systems. The Worldfip runs at 1 Mbps, permits a maximum of 250 nodes, uses a bus arbiter for arbitration, uses twisted pair cable and has application in real time control.

Detailed Description Text (53):
The Profibus is a Fieldbus network designed for deterministic communication between computers and PLCs. It is based on a real time capable asynchronous token bus principle. Profibus defines multi-master and master slave communication relations, with cyclic or a cyclic access, permitting transfer rates of up to 500 Kbps. The physical layer 1 (2-wire RS-485), the data link layer 2 and the application layer are standardized. Profibus distinguishes between confirmed and unconfirmed services, allowing process communication, broadcast and multitasking. The Profibus runs at 500 Kbps, permits a maximum of 127 nodes, uses token passing for bus arbitration, uses twisted pair cable and has application in inter-PLC communication.

Detailed Description Text (55):
The controller area network (CAN) is a serial bus that is designed to provide an efficient, reliable and very economical link between sensors and actuators. CAN uses

a twisted pair cable to communicate at speed of up to 1 Mbps with up to 40 devices. It was originally developed to simplify the wiring in automobiles but its use has spread to machines and factory automation products because of its useful features. Some of its features include the ability of any node to access the bus when the bus is quiet, non destructive bit wise arbitration to allow 100% use of bus bandwidth without loss of data, multimaster, peer to peer and multicast reception, automatic error detection, signaling and retries and data packets of 8 bit length. CAN is the basis of several sensor buses such as DeviceNET from Allen Bradley, CAN Application Layer from CAN in Automation or Honeywell's SDS. The CAN runs at 1 Mbps, uses CSMA for bus arbitration, uses twisted pair cable and has application in sensors and actuators.

Detailed Description Text (59):
The Parallel Intermodule (PI) bus uses the same basic structure as VMEbus but is adapted for real time, fault tolerant applications such as military mission critical systems. PI-bus is a synchronous, loosely coupled, message passing bus. A node may be master and slave capable or only slave capable. PI-bus uses the same backplane transceiver logic (BTL) interface as Futurebus+. PI-bus emphasizes fault tolerance and is inherently supportive of module level fault containment since it is a loosely coupled bus. It also contains features such as hardware supported intermodule communication containment boundaries, an error management protocol that supports determination of contaminated memory, the ability for software to control access to its memory and explicit software control of intermodule communication. PI-bus has no centralized control, the protocol uses a distributed vie for gaining bus mastership. The PI-bus is a 50 MBps bus using 32 parallel lines. Designers of PI-bus intended the bus operation to be a send and forget interface making it inappropriate as a real time interface in a tightly coupled architecture.

Detailed Description Text (75):
The RACEway bus is a proprietary bus uses the VME `P2` connector to access a crossbar switch to provide high speed concurrent data paths between boards in a VME chassis. It operates at speeds of 1280 Mbps using 32 parallel lines. The basic element of the RACEway is the RACE crossbar chip which has six I/O channels. A single crossbar chip can interconnect six nodes and provide up to three concurrent 1280 Mbps communication paths between node pairs. Topologies that can be created include fat-tree, switch ring and mesh. The RACEway is a preemtable circuit switched network. The RACEway uses a compelled protocol in that the receiving node can enforce flow control through the use of the 8-wire control and clocking signals. Data flow is bi-directional but can only go in one direction or the other at a time.

Detailed Description Text (81):
The Test and Maintenance bus is a linear, synchronous, multi-drop communication bus which transfers data between a master node and one or more slave nodes residing on a single backplane. It is used to communicate diagnostic control and status information between nodes. The TM-bus protocol supports up to 251 separate addresses plus the broadcast and multicast addresses. The TM-bus operates at speeds of 0.8 MBps using a serial line. Its intended application is for use with PI bus in military applications.

Detailed Description Text (103):
The Fiber Distributed Data Interface (FDDI) bus is a standard for a local area network with a transmission rate of 100 Mbps using a fiber optic cable as the transmission medium. FDDI implements a dual counter rotating ring topology and uses a token access method for packet transmission. FDDI is sometimes used as a higher speed backbone to interconnect several lower speed Ethernet networks. FDDI consists of three layers: Physical Layer, Medium Dependent (PMD), Physical Layer Protocol (PHY) and Media Access Control (MAC). The PMD layer specifies the digital baseboard point-to-point communication between two nodes in the FDDI ring. A ring consists of a set of nodes and unidirectional transmission medium segments connected in a serial closed loop configuration. The dual ring option consists of two identical ring configurations, where the secondary ring transmits in the reverse direction of the primary ring. The PHY Layer specifies the remaining aspects of the physical layer protocols. These protocols include decoding incoming data, encoding outgoing data and clock synchronization. The MAC Layer specifies fair and deterministic access to

the medium, address recognition and generation and verification of frames. Access to the ring is controlled by passing a token around the ring. If a node wants to transmit, it strips the token from the ring, transmits frames and then reinserts the token.

Detailed Description Text (126):
Surrounding the real-time kernel 154 is the operating system (OS) 155. The real-time kernel comprises the necessary operating system interface to allow it to execute on any desired operating system. The layer surrounding the OS includes various functional modules that perform various roles in the OpenBus system. These functional elements comprise a module for interfacing to sensors 158, I/O devices 160, motion related devices 162, computerized numerical control (CNC) devices 164, devices requiring motor control 166 and discrete I/O 168. In addition, functionality is provided to communicate with one or more networks 170. Further, a database module 172 provides the connectivity to a database that is used by the real-time kernel and application programs. A graphic module 74 provides graphics and drawing related functionality and an operator interface 156 provides the user interface for used by an operator of the system.

Detailed Description Text (129):
The real-time kernel in the target system (embodied in the embedded processor 22 in FIG. 4) dynamically changes in response to the structures and parameters defined by the user and represented in her/his application program 182.

Detailed Description Text (130):
With reference to FIG. 10 the development system environment of the present invention will now be described in more detail. As stated previously, the development system 180 comprises an application 182 provided by the user and a real-time compiler 184. The application comprises one or more event triggers 200, one or more event actions 202 and logic 204. These various elements combine to define the user's control program application.

Detailed Description Text (133):
The real-time kernel schedules the execution of event actions in accordance with the process state changes as reflected by the change entity value changes. Entities include but are not limited to variables, timers, counters and external input signals. These various entities are part of the program control logic making up the user's application. Any change to the value of an entity or any external signal triggers an immediate evaluation of the event trigger that incorporates that particular entity.

Detailed Description Text (134):
The programming logic functions as the basis for the event actions. The programming logic comprises pure logic, calculations, mathematical formulas, interfacing with sensors, discrete I/O, motion control, database operation, communication i.e., over networks and operator interface graphics.

Detailed Description Text (135):
All of the external information and programming logic defined by the user and embodied in her/his application comprises various elements such as event triggers, event actions, program variables, timers, counters, program logic, sensor information, motion trajectory planning, motion control, etc. All these elements are broken down, defined and represented via the frame p-code generated by the real-time compiler.

Detailed Description Text (136):
The p-code making up a frame is the smallest building block that enables the real-time compiler to generate code that executes with a response time required of a real-time system. The p-code making up frames comprises a precompiled, one step direct pointer to any piece of information or element which is required in order to perform the logic or operation of the frame. The logic or operation is performed on entities which include variables, timers, I/O port information, I/O values, etc. The precompiled direct pointer to the memory location of the particular entity permits rapid access to the values and references of the entities associated with a frame. These memory pointer references can be performed extremely rapidly with minimal

delay thus providing the real-time response needed by the application control program. This is in direct contrast to conventional compiled programming systems that typically involve run time memory address calculations, hash table calculations, heap and stack addressing, etc. in order to resolve memory references thus creating a huge overhead not present in the real-time kernel of the present invention.

Detailed Description Text (151):
Each entry in the event trigger table points, in turn, to an action. As stated previously, an action is comprised of one or more frames. For example, the event trigger entry 276 is shown to point to the action 300. In the example illustrated in FIG. 15, the action frame 300 comprises a plurality of frames 302. Each frame can be one of three different types. The three types of frames include: a program logic frame, an operation frame or a condition frame. A program logic frame includes such program logic as jumps, both conditional and unconditional, etc. Operation frames perform an action such as A=B * D, or assign a value to an entity or output signal. For example, a frame may assign a value to a timer, a variable or a digital or analog I/O port. A condition frame tests the given entities to be true or false, e.g., IF A=B+C THEN set an output signal.

Detailed Description Text (152):
A high level block diagram illustrating an illustrative example of a frame implementing an action is shown in FIG. 16. The action 310 comprises six frames with the first frame 312 being an operation type frame. Frame 312 assigns the value 10 to the variable entity `A`. Frame 314 assigns the value of variable `C` to `B`. The next frame 316 is a conditional type frame. If the value of variable `A` is greater than that of `B` control passes to frame 318 which outputs the value of variable `E` to an output port. Alternatively, control passes straight to frame 320 which is an operation type frame which assigns the value 2 to the variable `D`. The last frame 322 is a program logic type frame which performs a return function.

CLAIMS:

1. A control automation system for controlling a plurality of input and output (I/O) devices in accordance with a user application, said system connected to a network for communicating control automation information, said system comprising:

a development system coupled to said network, said development system for generating p-code, said p-code generated so as to embody event triggers, event actions and program logic that together function to implement said user application; and

at least one node controller coupled to said network for executing in real-time said p-code generated by said development system.

3. The system according to claim 1, wherein said development system comprises a real-time compiler for generating p-code in accordance with the event triggers, event actions and program logic of said user application.

7. A node controller apparatus for use in a control automation system, said system for controlling a plurality of input and output (I/O) devices in accordance with a user application, said system including a network for communicating control automation information, said apparatus comprising:

processor means for managing and controlling the operation of said node controller, said processor means for executing a real-time kernel, said kernel implementing said user application embodied in p-code;

network interface means for connecting said node controller to said network;

I/O interface means for connecting said node controller to said plurality of I/O devices; and

bus means for interconnecting together said processor means, said kernel means, said network interface means and said I/O interface means.

10. An apparatus for controlling a plurality of input and output (I/O) devices in accordance with a user application, said apparatus part of a control automation system, said apparatus comprising:

a processor operative to execute said user application;

I/O interface means for connecting said apparatus to said plurality of I/O devices;

kernel means adapted to communicate over a network and operative on said processor for:

reading, storing and determining changes to external input signals received from said plurality of I/O devices;

detecting changes to said external input signals and internal entities and for determining and resolving all event triggers corresponding to said detected changes;

marking all actions for execution that correspond to said event triggers that resolve true;

executing and implementing said actions marked for execution; and

determining any changes to values assigned to an entity making notification of said entity value changes.

12. In a computer system, a method of generating p-code for execution on a node controller as part of a control automation system for controlling a plurality of input and output (I/O) devices in accordance with a user application, said application including event triggers, event actions and program logic, said method comprising the steps of:

generating a plurality of pointer tables, each pointer table associated with either an external input signal or an entity, each pointer table comprising a plurality of pointer entries, each pointer entry pointing to an event trigger;

generating an event trigger table, said event trigger table comprising a plurality of event trigger entries, each event trigger entry corresponding to an action that references the particular external input signal or entity that points thereto;

generating a plurality of actions, each of said actions comprising at least one frame, said actions, said actions representing the generation of output signals and/or the modification of said internal entities; and

wherein said plurality of pointer tables, said event trigger table and said plurality of actions generated in accordance with said event triggers, event actions and program logic making up said user application.

13. A node controller apparatus for use in a control automation system, said system for controlling a plurality of input and output (I/O) devices, said system including a network for communicating control automation information, said apparatus comprising:

a processor for managing and controlling the operation of said node controller;

network interface means for connecting said node controller to said network;

I/O interface means for connecting said node controller to said plurality of I/O devices;

bus means for interconnecting together said processor means, said network interface means and said I/O interface means; and

kernel means adapted to communicate over said network and operative on said processor for:

reading, storing and determining changes to external input signals received from said plurality of I/O devices;

detecting changes to said external input signals and internal entities and for determining and resolving all event triggers corresponding to said detected changes;

marking all actions for execution that correspond to said event triggers that resolve true;

executing and implementing said actions marked for execution; and

determining any changes to values assigned to an entity marking notification of said entity value changes.

14. A node controller apparatus for use in a control automation system, said system for controlling a plurality of input and output (I/O) devices in accordance with a user application, said system including a network for communicating control automation information, said apparatus comprising:

processor means for managing and controlling the operation of said node controller, said processor means for executing a real-time kernel, said kernel implementing said user application embodied in p-code;

network interface means for connecting said node controller to said network; and

bus means for interconnecting together said processor means, said kernel means and said network interface means.

WEST

Generate Collection Print

L2: Entry 37 of 40 File: USPT Sep 10, 1996

DOCUMENT-IDENTIFIER: US 5555179 A
TITLE: Control method and control apparatus of factory automation system

Abstract Text (1):
A computer-implemented control method and a control apparatus of factory automation system having a cell formed by a plurality of working machines including at least one automated machine to execute a series of works in accordance with a control program. From a cell control program including specifications of working of entire cell that is input to the computer, information relating to working sequence of a plurality of working machines and information about control of inputs and outputs of a plurality of working machines are described, the specifications and informations are extracted by the computer. On the basis of the extracted information, a sequence control program by which control of the working sequence of a plurality of working machines and control of inputs and outputs of a plurality of working machines is generated by the computer are described. From the cell control program, information about operation control of the automated machine and information about synchronization of operation of the working machines are extracted by the computer. On the basis of the extracted information, an automated machine control program, by which processing for exercising control over positioning and operation trajectory of the automated machine are described, is generated by the computer. And working of the cell is controlled in accordance with the sequence control program and the automated machine control program.

Brief Summary Text (3):
As for the range indicated by the term "robot" herein, it refers to an automated machine automatically controlled in accordance with a previously taught program to conduct a complicated operation or work. Typically, the "robot" has a multi-axis mechanism. Industrial robots and numerical control machine tools are also comprehended in robots. Furthermore, the "cell" refers to a unit block of a working system for conducting a series of works, and the unit block includes an automated machine such as a robot and peripheral machines (or peripheral devices). Factory automation is typically formed by a combination of a plurality of cells individually having a specific work assigned thereto. A peripheral machine refers to a working machine or device including a conveyor, a parts feeding machine, an end effecter (including a mechanical hand) and so on. The peripheral machine operates in conjunction with an automated machine such as a robot in one cell. The peripheral machine (or device) is simpler in function and generally operated under sequence control.

Brief Summary Text (4):
In a conventional control method of an FA system, a sequencer (programmable controller) for controlling inputs and outputs (hereafter abbreviated to I/O) of various peripheral machines or devices and a robot controller for controlling robots are connected via a parallel I/O. Mainly, the sequencer always monitors signals carried over the I/O of various peripheral machines and the parallel I/O of the robot controller, synchronization of operations of various peripheral machines with operation of robots and exercises sequence control of the entire FA system.

Brief Summary Text (5):
As a known example relating to such a control method of the FA system, a control method of FA system described in JP-A-61-110204 can be mentioned. In this known example, there is described a method for describing programs of the robot controller, a visual device, and the sequencer by using a language having a common

unified system.

Brief Summary Text (6):
Typically in the above described conventional technique, robot controllers for controlling respective robots are prepared to control a plurality of robots included in the FA system, and each robot controller is taught a program for making a robot controlled by the robot controller perform one or more works. In this case, each robot must be generally taught a different program. In addition, when robots of different kinds are used, robots must be taught robot programs using robot languages of different systems in many cases.

Brief Summary Text (7):
As for a programming language of the sequencer for performing sequence control over peripheral machines (or devices) other than robots and the entire FA system, a latter diagram is the most popular language at present. However, this is a programming language having a form that is completely different from that of robot languages. Under these circumstances, the person in charge of constructing an FA system must learn several different programming languages. In addition, for controlling one FA system, a plurality of programs must be generated by using these different programming languages. This is one of major factors impeding the improvement in the development efficiency of the control program of the FA system.

Brief Summary Text (8):
As for the control method of FA system described in the aforementioned JP-A-61-110204 (corresponding to U.S. Pat. No. 4,730,258 issued on Mar. 8, 1988), a method of describing programs respectively for the robot controller, visual device, and sequencer by using a common language having one system is described. Thereby, it becomes unnecessary for the user to learn different programming languages. However, the program itself must be divided into and described as individual programs, each for individual processing such as robot control or sequence control so as to conform to the function of the device actually processing the individual program. (In the above described example of the known technique, such a unit program is referred to as task.) The user must produce a plurality of programs (tasks). In the same way as the foregoing, this results in a problem that the development efficiency of the control program of the FA system cannot be improved.

Brief Summary Text (9):
An object of the present invention is to provide a control method, and apparatus, whereby when a unit element (hereafter referred to as cell) including one set of a plurality of working machines is to be constructed to form an FA system, programs for controlling some works including synchronization of operations of the working machines included in the cell are unified as a cell control program, the user can describe directly the work specifications of the cell as a whole without being conscious of the configuration of the control device and without the necessity of producing separate programs by using different programming languages for respective control devices and learning several programming languages, and hence reduction of development man-hour and improvement of development efficiency can be achieved.

Brief Summary Text (11):
The present invention provides a control method and a control apparatus of a factory automation system using a computer having a cell formed by a plurality of working machines including an automated machine to execute a series of works in accordance with a control program.

Brief Summary Text (12):
In accordance with the present invention, the following control is performed. From a cell control program, in which specifications of the work of the cell as a whole is described, input to the computer, information relating to working sequence of a plurality of working machines and information about control of inputs and outputs of a plurality of working machines are extracted by the computer. The cell control program has information about a working sequence of a plurality of working machines, information about control of inputs and outputs of a plurality of working machines, information about operation control of the automated machine, and information about synchronization of operations of the working machines. On the basis of the extracted information, a sequence control program, in which control of the working sequence of

a plurality of working machines and control of inputs and outputs of a plurality of working machines are described, is generated by the computer. From the cell control program, information about operation control of the automated machine and information about synchronization of operations among the working machines are extracted by the computer. On the basis of the extracted information, an automated machine control program, in which processing for exercising control over positioning and operation trajectory of the automated machine is described, is generated by the computer. And working of the cell is controlled in accordance with the sequence control program and the automated machine control program.

Drawing Description Text (3):
FIG. 2 is a diagram showing an example of the terminal screen display made by a cell control program editing unit included in the apparatus of the present invention;

Drawing Description Text (10):
FIG. 9 is a diagram illustrating a method for generating a cell control program by using a data base of the module of operation sequence;

Drawing Description Text (11):
FIG. 10 is a diagram showing a cell control program obtained by encoding the Petri net shown in FIG. 3;

Drawing Description Text (14):
FIG. 13 is a diagram showing a code generation algorithm of cell control program conversion means for generating a sequence control program and a robot control program from a cell control program according to the present invention;

Drawing Description Text (15):
FIG. 14 is a diagram specifically illustrating a procedure for generating the sequence control program and the robot control program;

Drawing Description Text (16):
FIG. 15 is a diagram showing a surface language program for sequence control generated from the cell control program shown in FIG. 10 according to the present invention;

Drawing Description Text (17):
FIG. 16 is a diagram showing a surface language program for robot control generated from a cell control program shown in FIG. 10 according to the present invention;

Drawing Description Text (18):
FIG. 17 is a diagram showing a sequence control internal code program in Mnemonic expression generated from the surface language program for sequence control shown in FIG. 15 according to the present invention;

Drawing Description Text (19):
FIG. 18 is a diagram showing a robot control internal code program in Mnemonic expression generated from a surface language program for robot control shown in FIG. 16 according to the present invention;

Drawing Description Text (20):
FIG. 19 is a diagram showing the general configuration of the sequence control internal code program used in the apparatus of the present invention;

Drawing Description Text (21):
FIG. 20 is a diagram showing the general configuration of the robot control internal code program used in the apparatus of the present invention;

Drawing Description Text (22):
FIG. 21 is a diagram showing a processing flow of an interpreter of a sequence control internal code in a sequence control program interpretation and execution means included in the apparatus of the present invention;

Drawing Description Text (23):
FIG. 22 is a diagram showing a processing flow of an interpreter of a robot control

internal code in a robot control program interpretation and execution means included in the apparatus of the present invention;

Drawing Description Text (24):
FIG. 23 is a diagram showing the configuration of an example of interpretation and execution unit for a sequence control program and a robot control program shown in FIG. 1;

Drawing Description Text (25):
FIG. 24 is a diagram showing another example of interpretation and execution units for a sequence control program and a robot control program shown in FIG. 1;

Drawing Description Text (26):
FIG. 25 is a diagram showing an example of a sequence control program using a ladder diagram converted from the cell control program of FIG. 10 so as to correspond to the control apparatus of FIG. 24 according to the present invention;

Drawing Description Text (27):
FIG. 26 is a diagram showing an example of a robot control program converted from the cell control program of FIG. 10 so as to correspond to the control apparatus of FIG. 24 according to the present invention; and

Drawing Description Text (28):
FIG. 27 is a diagram showing another example of the robot control program converted from the cell control program of FIG. 10 so as to correspond to the control apparatus of FIG. 24 according to the present invention.

Detailed Description Text (3):
FIG. 1 shows an embodiment of a control apparatus for an FA system according to the present invention. In FIG. 1, numeral 10 denotes a cell including a robot 8 and various peripheral machines (or devices) 9a and 9b to form an FA system. Numeral 1 denotes a cell control program editing unit. The cell control program editing unit 1 is used by the operator on the screen of a display device to edit a cell control program 2 in a diagrammatic form using a Petri net. The cell control program 2 controls working of the robot 8 or the peripheral machines 9a and 9b forming the cell 10 and conducts state monitoring on the basis of transition rules of operation mode of the cell 10 as a whole.

Detailed Description Text (4):
Numeral 3 denotes a cell control program conversion unit formed by a microcomputer. This cell control program conversion unit 3 includes a cell control program storage 3a, a conversion procedure storage 3b, a computing unit 3c, a sequence control data storage 3d, an I/O control instruction train storage 3e, a sequence control program storage 3f, a robot control instruction storage 3g, a robot operation timing storage 3h, and a robot control program storage 3i. Each of the storages 3a, 3b, 3d, 3e, 3f, 3g, 3h and 3i is formed by a memory device. The computing unit 3c is a microcomputer CPU.

Detailed Description Text (5):
The computing unit 3c first stores the cell control program 2 in the cell control program storage 3a. Then in accordance with a conversion procedure stored in the conversion procedure storage 3b, the computing unit 3c separates and extracts sequence control data and an I/O control instructions from the cell control program 2 and stores them in the sequence control data storage 3d and the I/O control instruction storage 3e, respectively. On the basis of these types of extracted information, the computing unit 3c monitors I/O states of the robot 8 and the peripheral machines 9a and 9b. Concurrently, the computing unit 3c generates a sequence control program 4 which describes processing for performing control of operation sequence of machines included in the cell 10, i.e., the robot 8 and the peripheral machines 9a and 9b, and for controlling the I/O connected to these machines and processing for managing transition of the operation mode of entire cell 10. At the same time, the computing unit 3c separates and extracts robot control instructions and information of robot operation timing from the cell control program 2 and stores them in the robot control instruction storage 3g and the robot operation timing storage 3h, respectively. On the basis of these types of extracted

information, the computing unit 3c generates a robot control program 5 for performing positioning and trajectory control of the robot 8. The sequence control program 4 is interpreted and executed at high speed by a sequence control program interpretation and execution unit 6. In the same way, the robot control program 5 is interpreted and executed by a robot control program interpretation and execution unit 7.

Detailed Description Text (6):
FIG. 2 shows an example of screen display of a terminal in the cell control program editing unit 1 included in the apparatus of the present invention. The screen display forms a graphical user interface (GUI) having a multiwindow form.

Detailed Description Text (7):
The programming language of the cell control program 2 (hereafter referred to as cell control language) is one kind of graphic language, in which the operation sequence of machines in the cell 10 and the sequence of the operation mode of the entire cell 10 are described using a diagrammatic representation form of a Petri net. In this cell control language, the place of the Petri net is defined as the operation state of a machine or a device included in the cell 10, a transition is defined as a transition condition at the time of transition to the next operation state, and a directed arrow connecting between the place and the transition is defined as a direction of the sequence flow. The place in which a token is placed is defined as the activated state of the present time (hereafter referred to as an activated state). In each place, only one token can exist. Such a safe Petri net is considered as the cell control language.

Detailed Description Text (19):
By thus unifying the control program of the robot and the control program of the peripheral machines or devices as the cell control program 2 and describing the operation sequence of each unit so that the operator may understand it intuitively, the person developing the software of the FA system can easily produce the cell control program 2 so as to directly describe working specifications of the entire cell, without being conscious of any difference in configuration of the control apparatus of machines or devices in the cell and the programming language. That is to say, the developer of the cell control program 2 need not learn several different programming languages, unlike the conventional technique. As a result, the development efficiency of the program is improved and the maintenance of the program as software is also made easy.

Detailed Description Text (20):
FIG. 8 shows an example in which a module 18c is added to the Petri net shown in FIG. 3. A module is a partial control program. The operation sequence of the module 18c is an example of description of error processing conducted when insertion of the part 26 of the part insertion robot 19 has failed. The module 18c determines that when the center pusher 28 cannot lower to a predetermined height within a fixed time, insertion of the part 26 is considered to have failed for the same reasons and the part 26 is held again and exhausted.

Detailed Description Text (21):
In the cell control program editing unit 1 according to the present invention, a method of modification to add cell control program 2 can be made easily in the form of adding a module to the Petri net. Owing to such construction of the Petri net using modules, division of labor to make modules becomes possible in developing the cell control program 2. The cell control program editing unit 1 has a function of supporting such program development with modules. To be specific, program registration for each module is made possible. When developing the program for performing control over the entire cell 10, these modules are called occasion demands and linked on the net editing window 11, and the final cell control program 2 can thus be produced. If division of labor in developing the cell control program 2 is thus facilitated, the development efficiency of the cell control program can be improved.

Detailed Description Text (22):
An alternative method as described below can also be implemented. In constructing a cell of FA system, modules of operation sequences of general purpose units

frequently used are stored in the data base as ready-made modules. In newly constructing a cell, cell configuration data (information such as the types of units and specific I/O assignment) are input. Suitable modules according to the cell configuration are automatically called from the data base and linked to generate the final cell control program 2.

Detailed Description Text (23):
FIG. 9 is a diagram showing the outline of a method for generating the cell control program 2 by using the data base of modules of an operation sequence. In a data base 101 of operation sequence modules, modules of operation sequences of general purpose devices are stored. By inputting cell configuration data 102, such as kinds of units forming the cell (ARM 001, ARM 002, UNIT 001) and information (for example, the local argument x100 in the module corresponds to the actual port X100) of assignment of control signals of actuators and sensor signals of the units to I/O ports, necessary modules are automatically called from the data base 101. Called modules 103a and 103b are displayed on the screen of the cell control program editing unit 1 in a diagrammatic form of a Petri net. By connecting state places to be synchronized via a synchronizing place 103c, the modules 103a and 103b are linked and a cell control program 103 according to the cell configuration data 102 can be generated.

Detailed Description Text (24):
FIG. 10 shows the cell control program 2 obtained by encoding the Petri net shown in FIG. 3 which is edited by means of the cell control program editing unit 1. The cell control program editing unit 1 automatically encodes the edited Petri net by itself. In the cell control program 2 of FIG. 10, a net type 29 is a statement declaring whether the program described thereafter is a portion defining the operation sequence of units in the cell 10 (S-net: Sequence net) or a portion defining the transition rule of the operation mode of the cell 10 as a whole (M-net: Mode net). A mode number 30 is a statement declaring which operation mode includes the sequence of the program described thereafter, using a reference number of the operation mode.

Detailed Description Text (30):
FIG. 13 shows a code generation algorithm of the cell control program conversion unit 3. The code generation algorithm generates a sequence control program 6 and a robot control program 7 from the cell control program 2 as shown in FIG. 10. In the sequence control program 6, processing for controlling of operation sequence of the robots and various peripheral machines or devices (units) and for performing control over I/Os connected to these devices is described. In the robot control program 7, processing for control over positioning and operation trajectory of the robot is described. The cell control program 2 includes an S-net 38 which is a portion for defining the operation sequence of the units included in the cell 10 and an M-net 39 which is a portion for defining the transition rule of the operation mode of the cell 10 as a whole. Furthermore, each of the S-net 38 and the M-net 39 includes a "cell" block and a "def" block.

Detailed Description Text (31):
In the "cell" block, connection relations of respective states of the Petri net and operation of the units in respective states are defined. In the "def" block, end conditions of respective states are defined. The cell control program conversion unit 3 first extracts state connection data 41a from contents of the "cell" block of the M-net 39 and the "cell" block of the S-net 38 and extracts end condition data 41b of respective states from contents of the "def" block of the M-net 39 and the "def" block of the S-net 38. The state connection data 41a and the end condition data 41b are put together to form sequence control data 41. Furthermore, the cell control program conversion unit 3 extracts I/O control instructions 40 from the "cell" block of the M-net 39 and the "cell" block of the S-net 38, and first generates a surface language program for sequence control 44 from the I/O control instructions 40 and the above described sequence control data 41. Furthermore, this surface language program for sequence control 44 is converted to a sequence control internal code program 46 having a binary form suitable for interpretation and execution processing made by the interpreter (sequence control program interpretation and execution unit 6) in the controller at high speed. The surface language program for sequence control 44 and the sequence control internal code program 46 are named generically as sequence control program 4.

Detailed Description Text (33):
In the surface language program for sequence control 44, such a representation form that the user who sees it can intuitively understand contents is mainly intended. If occasion demands, program editing on this surface language level is also possible.

Detailed Description Text (34):
Furthermore, the cell control program conversion unit 3 extracts robot control instructions 42 from the "cell" block of the M-net 39 and the "cell" block of the S-net 38. Concurrently therewith, the cell control program conversion unit 3 extracts a reference number of the state in which these robot control instructions 42 are executed (hereafter referred to as state No.), i.e., information indicating robot operation timing 43. On the basis of these kinds of extracted information, the cell control program conversion unit 3 generates a surface language program for robot control 45.

Detailed Description Text (35):
Furthermore, the surface language program for robot control 45 is converted to a robot control internal code program 47 having a binary form suitable for interpretation and execution made by the interpreter (robot control program interpretation and execution unit 7) included in the controller.

Detailed Description Text (36):
The surface language program for robot control 45 and the robot control internal code program 47 are called collectively robot control program 5. In the same way as the surface language program for sequence control 44, the surface language program for robot control 45 is intended for easy understanding by the user. As occasion demands, program editing on the surface language level is also possible.

Detailed Description Text (37):
FIG. 14 specifically shows the procedure for generating the sequence control program 4 and the robot control program 5 on the basis of the algorithm shown in FIG. 13 by taking a part of the cell control program 2 of FIG. 10 as an example. In FIG. 14, 38a and 38b denote parts of the "cell" block and the def block of the S-net 38 of the cell control program 2, respectively. In the cell block 38a, a combination of a "cell" label 32 and a state transition expression 33 is state connection data 41a and 41a' of the Petri net.

Detailed Description Text (40):
From the state connection data 41a and the end condition data 41b, an antecedent portion 48a of a rule in the surface language program for sequence control 44 and a state transition instruction 48b (EXEC S104) of a consequent portion are generated (processing 1001 and 1002). That is, "if the state S103 ends normally and the synchronizing place REQ2 is satisfied" is generated as the antecedent portion 48a, and "the state S104 is executed (i.e., an activation flag and an execution flag of the state S104 are set)" is generated as the state transition instruction 48b.

Detailed Description Text (42):
At this time, from the cell label included in the state connection data 41a and 41a', state No. i.e., robot operation timing 43 and 43' are extracted simultaneously, and state label instructions included in the surface language program for robot control 45 are generated (processing 1005 and 1006).

Detailed Description Text (44):
Robot control instructions 42 are extracted from command block of the "cell" block 38a, and the instructions of the surface language program for robot control 45 are generated (processing 1008 and 1009).

Detailed Description Text (46):
Furthermore, in FIGS. 13 and 14, the I/O control instructions 40 is transferred entirely to the sequence control program 4. Alternatively, a part of these I/O control instructions 40 may be transferred to the robot control program 5.

Detailed Description Text (47):
In the code generation algorithm shown in the cell control conversion unit 3 of FIG.

13, the "cell" block consists of connection relations of states of the Petri net and unit operations in respective states and the "def" block consists of end conditions of respective states. Even if the cell control program 2 is described in a different form, however, code generation is possible in the same way. For example, even in such a description form that unit operations in respective states are collected into one block and connection relations and end conditions of respective states are collected into another block, a code generation algorithm similar to that of FIG. 13 is also applicable.

Detailed Description Text (49):
FIG. 15 shows the surface language program for sequence control 44 generated from the cell control program 2 shown in FIG. 10. FIG. 16 shows the surface language program for robot control 45. In processing of a rule 48 shown in FIG. 15, if state S201 ends normally and synchronizing place REQ1 ends normally, output Y200 and on-delay timer TD200 are set and the execution status of the state S202 is changed to executing (EXECS S202).

Detailed Description Text (51):
FIG. 17 shows a result obtained by converting the surface language program for sequence control 44 of FIG. 15 to the sequence control internal code program 46. FIG. 18 shows a result obtained by converting the surface language program for robot control 45 of FIG. 16 to the robot control internal code program 47.

Detailed Description Text (52):
Internal codes of FIGS. 17 and 18 are represented by converting binary codes to mnemonic codes. In the sequence control internal code program 46 of FIG. 17, instruction codes for conducting predetermined processing are described for each state of each unit in each operation mode. By a mode label 50, a unit label 51, and a state label 52, the start address of instruction codes in the program is specified. After the state label 52, instruction codes 53 is described. In the robot control internal code program 47 of FIG. 18 as well, the start address of internal codes for defining processing in each state is specified by such labels.

Detailed Description Text (53):
FIG. 19 and FIG. 20 show general configurations of the sequence control internal code program 46 and the robot control internal code program 47, respectively. With reference to FIG. 19, a label table 54 is located at the top of the program. The label table 54 defines program addresses of the mode label 50, the unit label 51, and the state label 52. In a mode block 55, processing relating to S-net 38 is described. The processing relating to S-net 38 is an operation sequence of each unit in the operation mode specified by the mode label 50. In a state block 56 included in the mode block 55, sequence processing having the state specified by the state label 52 as an activation condition (processing corresponding to the rule 48 in the surface language program 44) is described as instruction codes 53. In a regular processing block 57, contents to be processed regularly without regard to the operation mode (such as processing relating to M-net 39, timer or counter processing, interlock processing) are described.

Detailed Description Text (55):
FIG. 21 shows the processing flow of the internal code interpreter for sequence control in the sequence control program interpretation and execution unit 6 according to the present invention. First of all, the interpreter takes in current operation mode No. from a variable for storing operation mode No. (processing 1701) and jumps to the address of the mode label 50 indicating the current operation mode, included in the sequence control internal code program 46 (processing 1702).

Detailed Description Text (57):
On the basis of the state No. having the activation flag SA set, a jump is made to the state label 52 indicating the activated state included in the sequence control internal code program 46 (processing 1704). The instruction codes 53 described next to the state label 52 are interpreted and executed in order. That is to say, a jump is made to a function in which the actual contents of processing of instruction codes are defined (processing 1705). As many operands as need be are taken in from the sequence control internal code program 46 and processed (processing 1706).

Detailed Description Text (59):
If there is no following instruction code, i.e., if processing of the instruction code train 53 has been completed, then it is determined whether there is a different activated state (processing 1708). If there is a different activated state, a jump is made to the state label 52 indicating its activated state and the above described processing is repeated. If there is not another activated state, then a jump is made to the regular processing portion 57 of the sequence control internal code program 46 (processing 1709) and processing of the instruction codes 53 are conducted (processings 1710, 1711 and 1712).

Detailed Description Text (60):
If processing of regular processing block has been completed, then return to the beginning of this processing flow is made and the above described processing is repeated. By thus jumping to the state label of the activated state and processing only the instruction codes following the state label, it is possible to monitor only information to be monitored under the current activated state, i.e., only the external I/O signal and values of internal variables. It is thus possible to shorten the time required for processing of the sequence control internal code program 46.

Detailed Description Text (61):
FIG. 22 shows the processing flow of the internal code interpreter for robot control in the robot control program interpretation and execution unit 7 according to the present invention. Processing flow of FIG. 22 is the same as that of FIG. 21 except the portion relating to the regular processing portion 56.

Detailed Description Text (62):
FIG. 23 is a block diagram showing the configuration of a cell 10 including two robots 59 and 60 and peripheral machines or devices 61 and a controller 58 for controlling the cell 10, wherein the FA system control apparatus according to the present invention shown in FIG. 1 is applied. The controller 58 corresponds to the sequence control program interpretation and execution unit 6 and the robot control program interpretation and execution unit 7 shown in FIG. 1. The controller 58 and the cell control program conversion unit 3 (not shown in FIG. 23) may be disposed separately as shown in FIG. 23, or may be produced as one body. In this controller 58, a sequence control processor 58a and a robot control processor 58b are bus-connected by means of a shared memory 58c so as to exchange information closely between the sequence control processor 58a and the robot control processor 58b.

Detailed Description Text (64):
In a robot control program storage 58g, the robot control program 5 is stored. The robot control program 5 is interpreted and executed by an interpreter stored in an interpretation and execution procedure storage 58h. By a teaching unit 58e, taught position data and movement trajectory data of the robot are stored in a position storage 58f. To a sequence control processor 58a, I/Os (not illustrated) of various peripheral machines or devices 61 are connected via an I/O interface 58j. Various peripheral machines or devices are thus monitored and controlled.

Detailed Description Text (65):
In a sequence control program storage 58l, the sequence control program 4 is stored. The sequence control program 4 is interpreted and executed by an interpreter stored in an interpretation and execution procedure storage 58k. A communication unit 58d is used to communicate with a host computer connected to the controller 58. For example, the communication unit 58d is used when the sequence control program 4 and the robot control program 5 are to be transferred from a another computer on which the cell control program conversion unit 3 is mounted to the controller 58.

Detailed Description Text (69):
In case of such a configuration, synchronization of operations of respective units are attained and sequence of the entire cell 10 is controlled using simple on/off signals passing through parallel I/Os disposed between the sequencer 62 and the robot controllers 63 and 64. That is, signal lines output from the robot controllers are connected to input ports of the sequencer 62, and signal lines output from the sequencer 62 are connected to input ports of the robot controllers 63 and 64. In respective control programs, processing is conducted using these input/output signals as branch conditions of processing and synchronization of operations of

respective units is thus attained.

Detailed Description Text (70):
In this example, the cell control program conversion unit 3 has two functions. One of the functions is generating the sequence control program 4 described by using a programming language (surface language or internal codes) mounted on the sequencer 62 and the robot control programs 5 described by using a programming language (surface language or internal codes) mounted on the robot controllers 63 and 64 from the cell control program 1 described using a cell control language.

Detailed Description Text (71):
The other of the functions is automatically generating synchronizing processing using parallel I/Os as described above into the sequence control program 4 and the robot control programs 5 provided that connection information of parallel I/Os between the sequencer 62 and the robot controllers 63 and 64 is given beforehand.

Detailed Description Text (72):
Peripheral machines or devices (I/O) 65 and 66 connected to individual robot controllers 63 and 64 are controlled by I/O control commands in the robot control program 5 interpreted and executed in individual robot controllers 63 and 64.

Detailed Description Text (73):
FIG. 25 shows the sequence control program 4 using a ladder diagram obtained by converting the cell control program 2 of FIG. 10 so as to correspond to the control apparatus of FIG. 24. FIGS. 26 and 27 show the robot control program 5.

Detailed Description Text (76):
The robot control program 5 of FIG. 26 shows the program of ARM1. The robot control program 5 of FIG. 27 shows the program of ARM2. In case of ARM1, i.e., robot No. 1 59, for example, waiting state continues until the input No. 1 (I0), i.e., the output Y000 of the sequencer 62 turns on. If it turns on, processing proceeds to an operation instruction MOVP1(1) of the next step.

Detailed Description Text (77):
By thus automatically generating the program for synchronizing the sequencer 62 with the robot controllers 63 and 64, it becomes possible to cope with the control apparatus having a conventional configuration as shown in FIG. 24 and production of the cell control program 2 which is explicit in operation sequence is facilitated.

Detailed Description Text (78):
In the above described embodiments, the case in which a combination of a robot and other peripheral machines or devices forms the cell of the FA system has been described. In some cases, N.C. machine tools form the cell. In such a case, control can be performed in the same way. For example, processing for effecting working on predetermined parts using an N.C. machine tool may be described in the cell control program 2, and a part program for the N.C. machine tool may be generated by the cell control program conversion unit 3.

Detailed Description Text (79):
When the user constructs an FA system by combining a plurality of machines or devices including a robot, the above described embodiment of the present invention makes it possible to unify programs for controlling works of respective machines or devices as a cell control program and describe directly working specifications of the cell as a whole in the cell control program 2. It becomes unnecessary to produce separate programs using different programming languages for respective control devices as in the conventional technique. It also becomes unnecessary to learn several programming languages.

Detailed Description Text (80):
Furthermore, since the operation sequence of machines or devices in the FA system can be described structurally and explicitly, techniques of software engineering such as program addition, program modification, development with division of labor by using modules, and reuse of such modules can be applied.

Detailed Description Text (81):

Furthermore, since processing for synchronizing operation of a robot with operation of peripheral machines or devices can be described in such a form that the operator or user can intuitively understand at first sight, even a program for performing considerably complicated work can be described intelligibly. In addition, due to a synergistic effect, reduction of man-hours of program development and improvement of development efficiency can be achieved.

Detailed Description Text (82):
In turn, program productivity can be improved.

Detailed Description Text (83):
Furthermore, by using, as means for interpreting and executing the program thus produced, a method of advancing processing of internal codes on the basis of information of operation states of respective units included in the FA system, complicated work which is difficult to implement in the conventional method can be implemented.

Other Reference Publication (1):
"Petri Net Based Programming System for FMS" in IEICE Trans. Fundamentals vol. E75-A, No. 10, 1992, pp. 1326 to 1334.

CLAIMS:

1. A computer-implemented control method of factory automation system having a cell formed by a plurality of working machines including at least one automated machine to execute a series of works in accordance with a control program, said method comprising the steps of:

extracting, using a computer, information regarding a working sequence of said plurality of working machines and information regarding control of inputs and outputs of said plurality of working machines from a cell control program, which includes specifications of working of said cell as a whole, said cell control program being input to the computer, said cell control program further including information regarding a working sequence of said plurality of working machines, information regarding control of inputs and outputs of said plurality of working machines, information regarding operation control of said automated machine, and information regarding synchronization of operations of said working machines;

generating, on the basis of said extracted information by using the computer, a sequence control program in which control of the working sequence of said plurality of working machines and control of inputs and outputs of said plurality of working machines are described;

extracting, from said cell control program by using the computer, information regarding operation control of said automated machine and information regarding synchronization of operations of said working machines;

generating, on the basis of the information extracted in said extracting step using the computer, an automated machine control program in which processing for control over positioning and operation trajectory of said automated machine are described; and

controlling working of said cell in accordance with said sequence control program and said automated machine control program.

2. A control method of factory automation system according to claim 1, wherein

in said step of extracting information regarding the working sequence of said plurality of working machines, said information regarding the working sequence of said plurality of working machines comprises information regarding a relationship among operation states of a plurality of working machines and information regarding end conditions of said operation states;

in said step of generating the sequence control program, instruction of evaluation of operation state transition condition and execution of operation state transition

in said sequence control program are generated on the basis of said extracted information regarding working sequence of said plurality of working machines, and input/output control instruction codes of said plurality of working machines are generated on the basis of said information regarding control of inputs and outputs of said plurality of working machines;

said information regarding synchronization of operations of said working machines extracted from the cell program comprises a state reference number representing operation timing of said automated machine; and

said step of generating the automated machine control program comprises the steps of:

generating a state label instruction code of said automated machine control program on the basis of said state reference number, and

generating automated machine control instruction codes of said automated machine control program on the basis of said information regarding operation control of said automated machine.

3. A control method of factory automation system according to claim 1, wherein said cell control program is described by using a graphic language using a diagrammatic notation of a Petri net for representation of operation state transition of said plurality of working machines.

4. A control method of factory automation system according to claim 1, wherein said method further comprises the step of generating said cell control program, and

in said step of generating said cell control program, graphical user interface of multiwindow form is used, and in each window, operation sequences of said plurality of working machines, end conditions of respective states in the operation sequences, and operation contents in respective states are input and edited by cell control program editing means.

5. A control method of factory automation system according to claim 1, wherein said sequence control program and said automated machine control program, in which synchronizing processing between a sequence control processing system and an automated machine control processing system are described by using state label instruction code indicating such a state that operations of said working machines are currently being executed.

6. A computer-implemented control apparatus of factory automation system having a cell formed by a plurality of working machines including at least one automated machine to execute a series of works in accordance with a control program, comprising:

means for storing a cell control program input to the computer, said cell control program including specifications of working of said cell as a whole, said cell control program further including information regarding a working sequence of said plurality of working machines, information regarding control of inputs and outputs of said plurality of working machines, information regarding operation control of said automated machine, and information regarding synchronization operation of said working machines;

means for extracting information regarding a working sequence of said plurality of working machines and information regarding control of inputs and outputs of said plurality of working machines from said cell control program of said storing means;

means for generating, on the basis of the information extracted regarding the working sequence of said plurality of working machines and regarding control of inputs and outputs of said plurality of working machines, a sequence control program in which control of the working sequence of said plurality of working machines and control of inputs and outputs of said plurality of working machines are described;

means for extracting, from said cell control program, information regarding

operation control of said automated machine and information regarding synchronization of operations of said working machines;

means for generating, on the basis of the information extracted regarding operation control of said automated machine and information regarding synchronization of operations of said working machines, an automated machine control program in which processing for control over positioning and operation trajectory of said automated machine are described; and

means for controlling working of said cell in accordance with said sequence control program and said automated machine control program.

7. A control apparatus of factory automation system according to claim 6, wherein said cell control program is described by using a graphic language using a diagrammatic notation of a Petri net for representation of operation state transition of said plurality of working machines.

8. A control apparatus of factory automation system according to claim 7, comprising editing means for editing operation sequences of respective said working machines included in said cell control program, in diagrammatic notation of said Petri net.

9. A control apparatus of factory automation system according to claim 7, comprising editing means for editing operation sequences of respective ones of said working machines included in said cell control program by using instruction codes corresponding to diagrammatic notations of said Petri net in one-by-one correspondence.

10. A control apparatus of factory automation system according to claim 6, wherein said cell control program defines operation sequences of said plurality of working machines and transition rules of operation mode of said cell as a whole.

11. A control apparatus of factory automation system according to claim 6, wherein said cell control program is described by connecting, on the basis of information regarding synchronization of operations of said working machines, program modules of operation sequences respectively produced for said working machines.

12. A control apparatus of factory automation system according to claim 6, wherein said control apparatus further comprises:

cell control program editing means, said cell control program editing means including a data base for respectively, storing operation sequences of a plurality of working machines as ready-made program modules,

means for loading ready-made program modules from said data base in accordance with specifications input regarding said plurality of working machines, and

means for connecting said ready-made program modules on the basis of information regarding synchronization of operation of said working machines.

13. A control apparatus of factory automation system according to claim 6, wherein graphical user interface of multiwindow form produces said cell control program, said control apparatus further includes editing means, and in each window, operation sequences of respective said working machines, end conditions of respective states in the operation sequences, and operation contents in respective states are input and edited by said editing means.

14. A control apparatus of factory automation system according to claim 13, wherein said editing means internally represents said cell control program as instruction codes corresponding to all kinds of information input in said windows in one-by-one correspondence, and stores the instruction codes in one or more files.

15. A control apparatus of factory automation system according to claim 6, wherein said generating means for generating both said sequence control program and said automated machine control program generates said automated machine control program in the form of a surface language program which can be edited by the user, and

converts said surface language program to an internal code program to be executed by said means for controlling working of said cell.

16. A control apparatus of factory automation system according to claim 6, wherein said means for generating a sequence control program and said means for generating an automated machine control program generate said sequence control program and said automated machine control program in the form of internal code programs to be executed by said means for controlling working of said cell.

17. A control apparatus of factory automation system according to claim 6, wherein said sequence control program is described by using a programming language based upon description of rule having an IF-THEN form as a surface language.

18. A control apparatus of factory automation system according to claim 6, wherein said sequence control program and said automated machine control program in which synchronizing processing between a sequence control processing system and an automated machine control processing system is described by using state label instruction codes indicating such a state that operations of said working machines are currently being executed.

19. A control apparatus of factory automation system according to claim 6, wherein said cell working control means jumps directly to an address of a state label instruction code described in said sequence control program and selectively monitors only I/O signals and internal variables to be monitored in the state which is currently being executed in order to reduce the time required for execution processing of said sequence control program.

20. A control apparatus of factory automation system according to claim 6, wherein said cell working control means respectively comprise a sequencer and an automated machine controller mutually connected with I/O, and said generating means generates a program of ladder diagram for the sequencer and a program of a robot language for the automated machine controller.

21. A control apparatus of factory automation system according to claim 6, wherein said cell working control means comprises a controller having a sequence control processor, an automated machine control processor, and a shared memory connected to said both processors via bus lines, wherein said shared memory stores common information about operation states of said working machines used between said sequence control program and said automated machine control program.

WEST

Generate Collection Print

L2: Entry 38 of 40 File: USPT Jun 30, 1992

DOCUMENT-IDENTIFIER: US 5127090 A
TITLE: Map interface unit for industrial programmable logic controllers

Abstract Text (1):
"A Manufacturing Automation Protocol (MAP) interface unit couples a local area network (LAN) to a MAP network. The LAN includes a plurality of industrial programmable logic controllers (PLCs) and the MAP network includes a plurality of MAP devices. The MAP interface unit comprises a CPU for performing a plurality of routines. A first routine controls the coupling of the MAP interface unit to the MAP network and a second routine controls the coupling of the MAP interface unit to the LAN. A further routine disposed between the first coupling routine and the second coupling routine converts messages from the MAP devices on the MAP network to messages to the PLCs on the LAN and converts messages from the PLCs to messages to the MAP devices. These messages are either requests for data or responses with data to these requests."

Brief Summary Text (3):
Applicant's invention relates to networking of computer based devices and, more particularly, to an interface for communicatively coupling a local area network of programmable logic controllers to other computer based devices on a MAP (Manufacturing Automation Protocol) network.

Brief Summary Text (5):
This application is related to commonly assigned co-pending applications: Ser. No. 179,674 for "Peer-to-Peer Register Exchange Controller for PLCs"; and Ser. No. 258,779 for "Peer-to-Peer Register Exchange Controller for Industrial Programmable Controllers", the specifications of which are expressly incorporated herein.

Brief Summary Text (7):
As industrial automation advances, interconnectivity between various microprocessor based plant floor devices, such as programmable logic controllers ("PLCs"), and plant computers, becomes more and more desirable. However, as such microprocessor based devices and computers are made by various vendors utilizing their own various communication protocols, interconnectivity between such devices has been hampered, requiring application programs to be dedicated often to a single product and, consequently, non-transportable or reusable in future applications.

Brief Summary Text (8):
To help alleviate this problem, a seven-layer Open Systems Interconnection (OSI) communication model was specified by the International Standards Organization (ISO).

Brief Summary Text (11):
MAP is a communication networking standard initiated by General Motors Corporation and supported by the World Federation. It relies on the above seven independent, yet functionally supportive layers which serve as an accepted set of rules for data exchange within the manufacturing environment. The current MAP version is 3.0. MAP connectivity simplifies the task of data exchange between factory control devices and higher level manufacturing computers, typically supplied by a variety of vendors. So long as a device outputs its messages in accordance with MMS, all other devices on the MAP network should be able to recognize its messages.

Brief Summary Text (12):

Often, factory control devices, such as programmable logic controllers (PLCs) are interconnected via a local area network, or LAN, such as the SY/NET network of Square D Company, Palatine, Ill. It has been found to be beneficial to also permit these PLCs to exchange information with higher level computers.

Brief Summary Text (16):
It is an object of the invention to provide a Manufacturing Automation Protocol (MAP) interface for coupling a local area network (LAN) to a MAP network. The LAN includes a plurality of PLCs and the MAP network includes a plurality of MAP devices.

Detailed Description Text (6):
The modem 22 as provided by Concord includes a bus controller (not shown) which is connected to the controller board 24 to permit the controller board 24 to communicate with other devices on the MAP network 21.

Detailed Description Text (8):
The LAN card 30 can be a Square D Company Type SFI-5xx SY/LINK communications board which gives the controller box 40 direct access to the LAN 31 for communication with programmable controllers coupled thereto. In the present illustration, first and second PLCs 41,42, are shown coupled to the LAN 31; however many more such PLCs can be connected to the LAN 31. The LAN card 30 has two communication ports for cable connection and the board edge connector which is the communication link between the board and the controller box 40. The LAN card 30 includes a 9 pin, D-type female connector which allows the PLCs 41,42 on the LAN network to access the MAP network 21 via the LAN card 30.

Detailed Description Text (25):
The second option, edit configuration data, is of primary importance to the operation of the MIU 20 and permits adding and editing of configuration data in the configuration file 59. Specifically, through this option, one enters configuration data about all possible connections and naming conventions of devices and variables on both the MAP network 21 (MAP configuration data) as well as the LAN 31 (LAN configuration data). The configuration data includes LAN configuration data identifying devices coupled to the LAN 31, and MAP configuration data identifying devices coupled to the MAP network 21. The LAN configuration data includes a list of all of the programmable logic controllers connected to the LAN 31, which in the illustrated case comprises the PLCs 41,42. A unique identifying name is assigned to each of the PLCs 41,42, and variables are assigned for each. In addition, each variable is assigned read/write protection, and MAP nodes are specified having access to these variables. A unique number is assigned for each of the PLCs 41,42 as well as the address of each of the PLCs 41,42 on the LAN 31. A first series of three digits is automatically placed on the address, representing the MAP address of the MIU 20 on the MAP network 21. In addition, a name can be assigned to a programmable controller variable, for example a bit of a register. These are referenced as named variables by a communicating MAP node when reading or writing to a register of one of the PLCs 41,42. In the example above wherein the computer 44 was requesting a temperature read with such information being stored in the PLC 42, the respective register of the PLC 42 would be assigned the name "temperature". Thus when a temperature read request is received by the MIU 20 from the computer 44, the MIU 20, in accordance with the configuration data, recognizes this as a request for the data in the respective assigned register of the PLC 42.

Detailed Description Text (26):
Similarly, MAP configuration data is also established. This MAP configuration data identifies each of the MAP nodes with which any of the PLCs 41,42 desires to connect and communicate with.

Detailed Description Text (27):
The Mapware routine 54 is provided with the Concord modem. Signals on the MAP network 21 are converted by the Concord modem into a form recognized by the controller board 24, and vice versa. In effect, the Mapware routine 54 builds layers 1-6 of the MAP communication protocol, and the MIU routine 56 provides layer 7, that of deciphering and decoding the application layer.

CLAIMS:

1. A Manufacturing Automation Protocol (MAP) interface unit for coupling a local area network (LAN), to a MAP network for the transmission of messages or data there between, said LAN including a plurality of programmable logic controllers (PLC) and said MAP network including a plurality of MAP devices, the MAP interface unit comprising:

(a) first coupling means coupled to the MAP network for transmitting messages or data from said MAP interface unit to said MAP network;

(b) second coupling means coupled to the LAN for transmitting messages or data from said MAP interface unit to said LAN;

(c) means disposed and connected between said first coupling means and said second coupling means for converting a MAP message or data from a respective one of said MAP devices on said MAP network into a LAN message for transmission to a respective one of said PLCs and for converting a LAN message or data from said respective one of said PLCs connected on said LAN network into a MAP message for transmission to a respective one of said MAP devices; and

(d) wherein said MAP messages and said LAN messages comprise requests for data and responses with data between said PLCs and said MAP devices.

4. A Manufacturing Automation Protocol (MAP) interface unit for coupling a local area network (LAN) including a plurality of programmable logic controllers (PLCs) connected thereto to a MAP network for the transmission of messages or data therebetween including a plurality of MAP devices connected thereto, the MAP interface unit comprising:

(a) a first coupling means connected to the MAP network for transmitting messages or data from said MAP interface unit to said MAP network;

(b) a second coupling means connected to the LAN for transmitting messages or data from said MAP interface unit to said LAN;

(c) a controller board means connected between the first and second coupling means for performing a plurality of routines, said plurality of routines including a MAP interface routine and a LAN interface routine, and responsive to messages and data from either the LAN or MAP network for controlling the transmission of messages and data through said first and second coupling means;

a MAP request queue means connected to the MAP unit and accessible by said MAP interface routine and said LAN interface routine running in said controller for storing a MAP request consisting of a request for data from a respective one of said PLCs by a respective one of said MAP devices and received from said MAP interface routine;

(e) means for determining a corresponding LAN address of said respective PLC;

(f) means for sending by said LAN interface routine said stored MAP request to said corresponding LAN address;

(g) a MAP response queue means connected to the controller and accessible by said MAP interface routine and said LAN interface routine, for storing a MAP response generated in response to said sent MAP request by said respective PLC;

(h) means for receiving by said LAN interface routine said MAP response and storing said MAP response in said MAP response queue;

(i) means for determining a corresponding MAP address of said MAP device of said stored MAP response; and

(j) means for sending by said MAP interface routine said MAP response from said MAP response queue to said corresponding MAP address.

9. A Manufacturing Automation Protocol (MAP) interface unit for coupling a local area net (LAN) including a plurality of programmable logic controllers (PLCs) to a MAP network including a plurality of MAP devices for transmission of data and messages therebetween, the MAP interface unit comprising:

(a) first coupling means for coupling said MAP interface unit to said MAP network;

(b) a second coupling means for coupling said MAP interface unit to said LAN;

(c) a controller board connected between the first and second coupling means for performing a plurality of routines, said plurality of routines including a MAP interface routine for controlling said first coupling means and a LAN interface routine for controlling said second coupling means;

(d) means connected to the controller board for storing MAP request data received from said MAP interface routine, said MAP request data consisting of a request for data from a respective one of said PLCs by a respective one of said MAP devices and received from said MAP interface routine;

(e) means for determining a corresponding LAN address of said respective PLCs;

(f) means responsive to said LAN interface routine for sending said stored MAP request to said corresponding LAN address;

(g) a MAP response queue means responsive to said MAP interface routine and said LAN interface routine for sending a MAP response generated in response to said sent MAP request by said respective PLCs;

(h) means for receiving data from said LAN interface routine said MAP response and for storing said MAP response in said MAP response queue;

(i) means for determining a corresponding MAP address of said respective MAP device of said stored MAP response;

(j) means responsive to data generated by said MAP interface routine for sending said MAP response from said MAP response queue to said corresponding MAP address;

(k) means for storing a LAN request generated by said LAN interface routine, said LAN request consisting of a request for data from a respective one of said MAP devices by a respective one of said PLCs;

(l) means for determining a corresponding MAP address of said respective MAP device;

(m) means for sending said stored LAN request to said corresponding MAP address;

(n) a LAN response queue means connected to the controller and responsive to data generated by said MAP interface routine and said LAN interface routine;

(o) means for sending a LAN response generated in response to said sent LAN request by said respective MAP device to said LAN response queue;

(p) means for determining a corresponding LAN address of said respective PLC of said stored LAN response; and

(q) means for directing said LAN response from said LAN response queue to said corresponding LAN address.

WEST

End of Result Set

☐ Generate Collection Print

L2: Entry 40 of 40 File: EPAB Feb 19, 2003

DOCUMENT-IDENTIFIER: EP 1284446 A1
TITLE: Industrial controller automation interface

Abstract Text (1):
CHG DATE=20030403 STATUS=N>ayerayerayerayerAn automation interface is provided for interacting with industrial controllers. The automation interface provides for programming, editing, monitoring and maintenance of industrial controllers programmatically from a local or remote location. The automation interface component is adapted to communicate with industrial controllers by integrating a computer process interface library into the automation interface component. The computer process interface library exposes the automation interface component to client application processes, so that the client application processes can communicate with the at least one industrial controller programmatically. The automation interface is provided with functionality for downloading, uploading and programming of control programs to the processors of the industrial controllers. irst